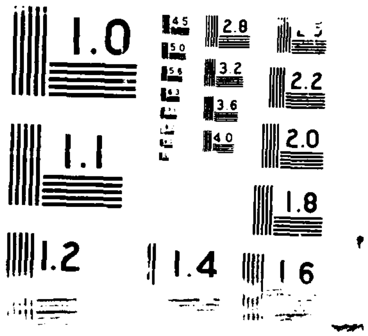
AD-A198 130 ALGORITHMS FOR ROBUST IDENTIFICATION AND CONTROL OF LARGE SPACE STRUCTURE (U) BUSINESS AND TECHNOLOGICAL SYSTEMS INC LAUREL MD J V CARROLL 14 MAY 88 1/1

UNCLASSIFIED BTS63-88-34/AB AFOSR-TR-88-0755 F/G 22/1 NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.2
1.4
1.6

2

# ALGORITHMS FOR ROBUST IDENTIFICATION AND CONTROL OF LARGE SPACE STRUCTURES

## PHASE I FINAL REPORT

by

BTS

2

BTS63-38-34/ab
J1131

May 1988

# ALGORITHMS FOR ROBUST IDENTIFICATION AND CONTROL OF LARGE SPACE STRUCTURES

## PHASE I
## FINAL REPORT

by
James V. Carroll

Prepared for
U.S. AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Under
Contract No. F49620-87-C-0099

Sponsored by

SDIO/INNOVATIVE SCIENCE AND TECHNOLOGY OFFICE

Prepared by
BUSINESS AND TECHNOLOGICAL SYSTEMS, INC.
14504 Greenview Drive, Suite 500
Laurel, MD 20708

The views and conclusions contained in this document are those of the author(s) and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U. S. Government.

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>UNLIMITED |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>J1131 | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>AFOSR-TR- 88-0755 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Business and Technological Systems, Inc. — 6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>Air Force Office of Scientific Research |
| 6c. ADDRESS (City, State and ZIP Code)<br>14504 Greenview Drive, Suite 500<br>Laurel, MD 20708 | 7b. ADDRESS (City, State and ZIP Code)<br>Buiding 410<br>Bolling AFB, DC 20332-6448 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>SIDO — 8b. OFFICE SYMBOL (If applicable) NA | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>F49620-87-C-0099 |
| 8c. ADDRESS (City, State and ZIP Code)<br>Innovative Science & Tech. Office<br>1717 H Street<br>Washington, DC 20301-7100 | 10. SOURCE OF FUNDING NOS.<br>PROGRAM ELEMENT NO. 63222C — PROJECT NO. K832 — TASK NO. F7 — WORK UNIT NO |

11. TITLE (Include Security Classification)
Algorithms for Robust Identification and Control of Large Space Structures

12. PERSONAL AUTHOR(S)
James V. Carroll

| | | | |
|---|---|---|---|
| 13a. TYPE OF REPORT<br>Final | 13b. TIME COVERED<br>FROM 8/87 TO 3/88 | 14. DATE OF REPORT (Yr., Mo., Day)<br>1988 MAY 14 | 15. PAGE COUNT<br>78 |

16. SUPPLEMENTARY NOTATION

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB. GR. | Large Space Structures, Robust Control System Identification, Computational Algorithms |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A new method of providing robust attitude control for tracking and slewing maneuvers for flexible structures in orbit is developed, and preliminary analyses and performance studies are conducted. The key elements of the method are system identification inreal time, based on canonical variate analysis, and adaptive robust control using Model Predictive Control. The CVA method also possesses the built-in capability for performing statistically optimal model order reduction. Ccmputational algorithms are developed using several low order flexible models. The results of this Phase I SBIR feasibility effort demonstrate that the new method is subject to careful design to reduce computer core size problems, but that its overall performance offers encouraging potential for more complete development

| | | |
|---|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐ | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED | |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Dr. Anthony K. Amos | 22b. TELEPHONE NUMBER (Include Area Code)<br>(202) 767-4937 | 22c. OFFICE SYMBOL<br>AFOSR/NA |

EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

# PREFACE

The work performed under Contract F49620-87-C-0099 was sponsored by the U.S. Air Force Office of Scientific Research in support of the Strategic Defense Initiative Organization. The program technical monitor was Dr. Anthony K. Amos. The principal investigator at Business and Technological Systems, Inc. was Dr. James V. Carroll, and Professor Mark J. Balas of the University of Colorado at Boulder was a consultant. Sections of this report were authored by John Garner of Business and Technological Systems, Inc., and Dr. Fred Austin of Grumman Aerospace Corp.

# Contents

# List of Figures

# List of Tables

# 1 IDENTIFICATION AND SIGNIFICANCE OF THE PROBLEM

Robust control is a method of maximally enhancing stability and performance of a dynamic system in the presence of modeling and environmental uncertainties. It is natural to consider using robust control methods to improve the dynamic behavior of Large Space Structures (LSS).

A key aspect of robust control is adaptability. Adaptability is enhanced when there exists a mechanism for updating the control dynamic model either on a periodic and/or an as-needed basis. The need for internal, automated model updates becomes more acute when either the dynamic plant is difficult to model or when its operating environment is either poorly defined or conducive to causing changes in the dynamics of the physical plant.

These conditions apply to the LSS in orbit. In the first place, models of space structures which sufficiently describe the dynamics are quite often of unrealistically high system order. Researchers are now analyzing ways to meaningfully utilize models of order 10,000. A great deal of research has been devoted to model approximation schemes, and it will likely remain a major activity well into the future, as many schemes are application dependent. A related problem is that analytic models are hard to verify on hardware on the 1-g Earth. It appears that a scheme consisting of online identification plus a controller which adapts optimally to updated dynamic information is a meaningful basis for highly robust, active LSS control.

The crux of the problem is that LSS dynamics are essentially infinite-dimensional, best described by the mathematics of Hilbert spaces and partial differential equations. In addition, classic and/or reliable control synthesis methods generally require finite-dimensional, linear models (Balas, 1982). Further, the realities of processing throughput requirements, not to mention software and hardware issues of reliability, redundancy management, etc., create a strong need for using control design models[1] whose system order is very judiciously selected.

Finding an appropriate scheme (or family of schemes) for LSS control is becoming an increasingly critical issue as national policy moves toward deployment of these structures in space for many purposes, including scientific research, communication, manufacturing, manned life support, and defense. Defense and scientific requirements in particular place very tight performance requirements on such metrics as pointing accuracy. Pointing accuracy is even more difficult to achieve due to one effect of the economics of space operations: lighter, more flexible and larger structures. Effective control of slewing, pointing and other LSS attitude maneuvers clearly implies, then, control of structural flexibility.

Typical LSS performance metrics - for example, beam or telescope line-of-sight

[1] For the purposes and scope of this report, three classes of LSS models are referred to in the text: (1) "truth" models - reasonably accurate simulation models of an actual LSS structure; (2) control design models - of lesser order, used in control synthesis; (3) on-board models - part of the flight software, typically in the controller, estimator and/or identifier subsystems. Often, but not necessarily, on-board models are equivalent to control design models. Model types (2) and (3) are usually linear, and (1) may be nonlinear.

(LOS) accuracy (ranging near 1 $\mu$rad), modal damping, surface shape errors (antenna applications), etc. - can be highly sensitive to changes in the LSS model, as well as to sensor measurement and to sensor and actuator location. Another performance issue not usually very critical in the case of design exercises operating on simpler dynamic models (e.g., rigid body) concerns the dynamic interactions of LSS sensors and actuators with the main structure. These interactions arise not only from the nature of these devices, but also from their location on the LSS.

The point of this discussion is that the LSS control design problem, of its nature, has extra dimensions of interrelationship among the modeling, dynamics and control synthesis disciplines, than are found in most other control design problems.

Another implication of the discussions above is that "simple" (ie, low order) dynamic models are traditionally nearly always appropriate for initiating a design effort or establishing proof of concept to a control design idea. This approach is undertaken at much greater risk in the case of LSS control design. The so-called spill-over effects which arise from the dynamic interaction of unmodeled with modeled modes pressure the designer to use a higher order design model, while design algorithmic, software development, processing hardware, system throughput and reliability issues, etc., exert strong, opposite pressures. This is a classic tradeoff, but it is very amplified in the LSS problem. Reasonable LSS dynamic models usually have far too many modes to admit effective control design and implementation.

Briefly, modeling, dynamics and control are highly interrelated design disciplines. Control design is driven primarily by spacecraft dynamic characteristics, performance requirements, type and degree of external and internal disturbances, and type, number and availability of sensors and actuators. The elastic or bending modes are theoretically infinite in number, and typically have low natural damping. Performance criteria must be well defined, and all disturbances must also be well defined and modeled, in order to have a well-posed control problem. Limitations on control bandwidth dictate a control design based on a reduced order model (ROM: the so-called "control design" model), which contains only selected low frequency modes. A critical design issue to be addressed is efficient identification and selection oi these ROM modes.

The key low frequency modes usually require the most control effort due to their larger settling time, and to their larger effect on performance degradation. However, the design method ought not allow the more poorly known high frequency modes to go unstable.

The Phase I effort of this work has explored the feasibility of applying robust identification and control techniques to the LSS control problem. A more detailed description of these techniques, and also of their application to some relatively small albeit representative LSS models, is provided in later sections. In the following, refer to Figure 1.

The *identification method* applies a canonical variate analysis (CVA) technique to selected input and measurement signals of the process to be identified, and thereby extracts model parameters. Model order is statistically determined using built-in criteria. By using such computationally stable algorithms as singular value decomposition, CVA identification is well-suited to LSS applications. In fact, its major applications success

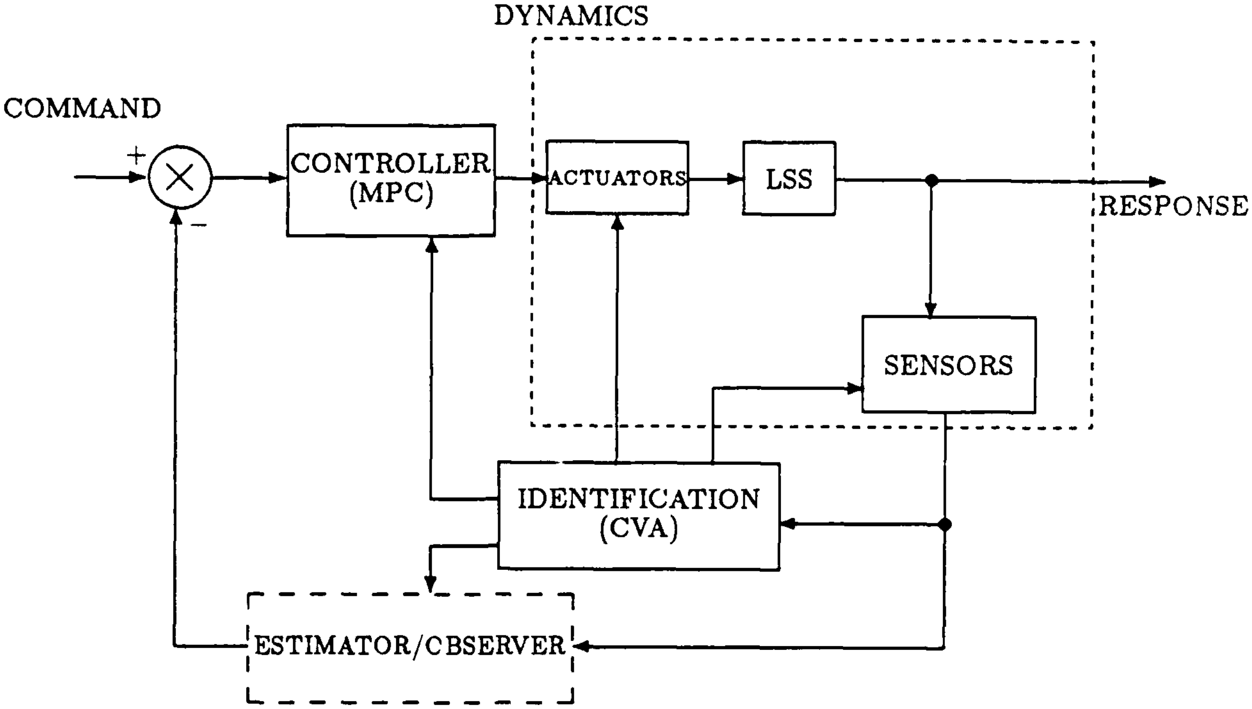


Figure 1: LSSICS Overview, Showing Functional Relationship of Modeling, Dynamics and Control

to date is in the very related area of flutter suppression (Larimore and Mehra, 1985), in which real time identifications were performed successfully on a full model, 1/4 scale F-16 with wing stores, in a wind tunnel at NASA/Langley. This was accomplished in a complete "hands off" mode. CVA identification is discussed in more detail in Section 2.2.

The *control approach*, called Model Predictive Control (MPC), is a multi-input/multi-output (MIMO) output controller, capable of driving several outputs simultaneously to their individual (input commanded) setpoints, or of tracking an input reference trajectory. The control needed to achieve the desired output, or response, values is obtained by comparing the difference between a "desired" output reference trajectory and the predicted no-input trajectory, against the predicted control response trajectory. The control equation can be converted into an overdetermined system, and solved for the control inputs using a weighted minimization criterion. These weights become control design parameters, in a manner similar to weighted linear quadratic (LQ) design. Other design parameters relate to shaping the output trajectory and to the length of the control prediction window. See Section 2.3 for more detail.

Most standard control design methods have difficulty in the LSS arena because they work best on single-input-single-output (SISO) systems, or if MIMO, have difficulty in dealing with spill-over dynamics, failed actuators, changing dynamics, and control energy limits. The model predictive control scheme proposed here as a viable, (logically) flexible, robust MIMO method is able to address all of these concerns as an implicit feature of its design process. MPC by itself is robust; it is therefore protected from some of the stability concerns which can afflict LQ-type designs when the dynamic plant is not perfectly known. Such stability concerns are usually very serious in LSS applications, where one typically must neglect numerous *known* modes, in addition to being forced to neglect unknown modes.

MPC's inherent robustness enhances overall stability in the presence of environmental uncertainties or failed subsystems. System performance, however, is affected by the accuracy of the on-board model. An on-board identification scheme offers the possibility to maintain good models, thereby positively impacting both performance and robustness. That is to say, maintenance of an accurate model is maximally robust, as more sudden degradations can be tolerated, until system identification determines the best "new" model.

At issue with identification algorithms is their computational stability, efficiency and overall reliability. The method proposed here, which is based on canonical variate analysis, scores exceptionally well on these issues, has been proven in related applications, and is thus a natural candida.e for LSS applications.

Inclusion of an identification module as a key control subsystem (Figure 1) is recognition of the tight interdependence among modeling, dynamics and control, as discussed above. The estimator shown in Figure 1 is not considered to be within the scope of the research effort, but is recognized as being very important in an actual mechanization. It is assumed to be a Kalman filter or a Luenberger-type reduced order observer. Together these modules comprise the Large Space Structure Identification and Control System (LSSICS). Other important modules, such as Failure Detection and Isolation (FDI), are not shown for purposes of clarity.

LSSICS has the following attributes relative to crucial LSS control design issues:

1. Identification

The CVA method of system identification has been recently applied to a variety of systems. It has a number of features that are well suited to real-time as well as offline identification on microprocessors including:

a) General state space model including inputs, and process and measurement noise

b) Multi-input multi-output systems

c) Computation using the singular value decomposition

d) Numerically stable and accurate - never fails

e) Automatic determination of the best choice of model state order

f) Finite amount of computation - nonrecursive

g) No initialization required - accurate on small samples

h) Near the maximum likelihood lower bound in accuracy

i) Discrimination of closely spaced spectral peaks

j) Simultaneous identification of transfer function and noise spectrum due to disturbance process

These properties of the algorithm give it a unique place among the other currently available algorithms. It is reliable, accurate, and yet will handle the very difficult problem of system identification of multivariable systems of high order using short data lengths.

2. Control

The proposed Model Predictive Control method:

a) Uses state space form of linear models, either as impulse or step response functions

b) Is an output predictive controller. There is thus no intrinsic need for full state feedback, full order observers, etc.

c) Like CVA, is structured to operate directly on MIMO systems

d) Uses linear system model in computing control inputs

e) Has demonstrated robustness to model uncertainties and/or changes in the dynamic properties of the system being controlled.....

Phase I of this effort involved demonstrating the feasibility of applying LSSICS to the LSS problem. The results obtained to date are presented in Section 4, and provide further background on the overall LSSICS techniques. The main objective of follow-on Phase II work is to further refine and tune LSSICS, integrate its identification and control modules into one efficient software entity, demonstrate its capability on "realistic" LSS simulation models, compare its overall performance with current alternate methods, and develop on-board software implementation requirements.

# 2 MATHEMATICAL FORMULATION

The basic objective of the Phase I research is to demonstrate, via simulation, the feasibility of applying MPC control and CVA identification to the design of robust controllers for flexible space structures. Lesser attention has been given to other key elements of a fully operational system, for example the Fault Detection and Isolation (FDI) element. The basic objective was achieved by defining an appropriate baseline Large Space Structure (LSS) model and environment, and establishing test cases for performance analysis. The MPC and CVA algorithms were verified for this new application, and modular simulation tests were run both for design and performance. In this section, mathematical background and formulation of the LSS model and dynamic process is presented, as well as salient features of the CVA and MPC algorithms.

## 2.1 LSS Dynamic Model

The LSS is a distributed parameter system - consequently, it is infinitely dimensional in theory, and potentially very large in practice (analysts are working now with models containing 10,000 modes!). However, implementable controllers are difficult to obtain for a distributed parameter system (Balas, 1982), a fact which also applies to Model Predictive Control. For this reason, LSS dynamics are usually converted to a reduced order model for control synthesis especially, but also for verification purposes as a "truth" model. We now summarize briefly the development of the basic structural dynamic equations.

A generic structure could be described as a continuum via the following partial differential equation:

$$m(x,y,z)u_{tt}(x,y,z,t) + D_o u_t(x,y,z,t) \ + \ A_o u(x,y,z,t) \ = \ F(x,y,z,t) \qquad (1)$$

where $u(x,y,z,t)$ is a function of displacements of the structure from its equilibrium position, which results from the applied force distribution $F(x,y,z,t)$ - also a vector - and from unmodeled and transient disturbances; also, $D_o$ and $A_o$ are linear operators in $x,y,z$ and $m(x,y,z)$ is the (positive) mass density measure. The force distribution vector and displacement vector may include torques and rotations, respectively.

The most popular method in structural analysis for converting Equation (1) to a reduced order model is the *finite element method* ( ref's III.3 to III.5 in Balas, 1982 ). A set of finite elements or patch functions $\theta_k(x)$ is used in approximating the solution to Equation (1) by

$$u(x,t) \ = \ \sum_{k=1}^{No} \hat{q}_k(t)\theta_k(x) \qquad (2)$$

for structural members homogenous in the $(y,z)$ directions, where the $\hat{q}_k$ are chosen to minimize the mean square equation error when (2) is substituted into Equation (1), and $N_0$ is the number of modes used in the reduced order model. System order $N$ is thus $2N_0$. The $\theta_k(x)$ may be linear combinations of piecewise linear functions, cubic splines or the actual LSS mode shapes, when the latter are known.

Defining displacement coordinates $\hat{q}(t) = [\hat{q}_1, ..., \hat{q}_{N_0}]^T$, Equation (1) becomes

$$M\ddot{q} + D\dot{q} + Kq = B^o f \tag{3}$$

The displacement equation (3) is readily converted into (linear) state space form for CVA and MPC use; however, it is common for computational efficiency and stability to convert Equation (3) to *modal* form via the orthogonal transformation

$$q = Uu$$

resulting in

$$\ddot{u} + \tilde{D}\dot{u} + \Lambda u = \tilde{B}f \tag{4}$$

Hence

$$\dot{x} = Ax + Bf \tag{5}$$

is the final desired state space form, where $x = [u^T \dot{u}^T]^T$ and

$$A = \begin{bmatrix} 0 & I_N \\ -\Lambda & -\tilde{D} \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ \tilde{B} \end{bmatrix}$$

and, considering also Equations (3) and (4), we have the orthogonal transformations (defining $U$)

$$U^T MU = I_N, \ U^T KU = \Lambda;$$

also

$$\tilde{B} = U^T B^o \ , \ \tilde{D} = U^T DU$$

and, finally, $M, D, K$, and $B^o$ in Equation (3) are, respectively, mass, damping, stiffness, and control influence matrices. There is a companion observation equation to Equation (5),

$$y = Cx \tag{6}$$

As formulated, Equations (5) and (6) are the set of control model equations used in MPC design. *Model error vectors* representing disturbances, unmodeled modes, biases, etc., are unmodeled in the controller design equations but generally used additively in the "truth" model equivalent to Equations (5) and (6). Structural analysis programs such as NASTRAN are generally used to generate the reduced order model.

It is very important that the models derived by the methods cited above be sufficiently accurate. This fact is critical to controlling the LSS, and it also motivates the need for stable, reliable on-board identification (see Section 2.2).

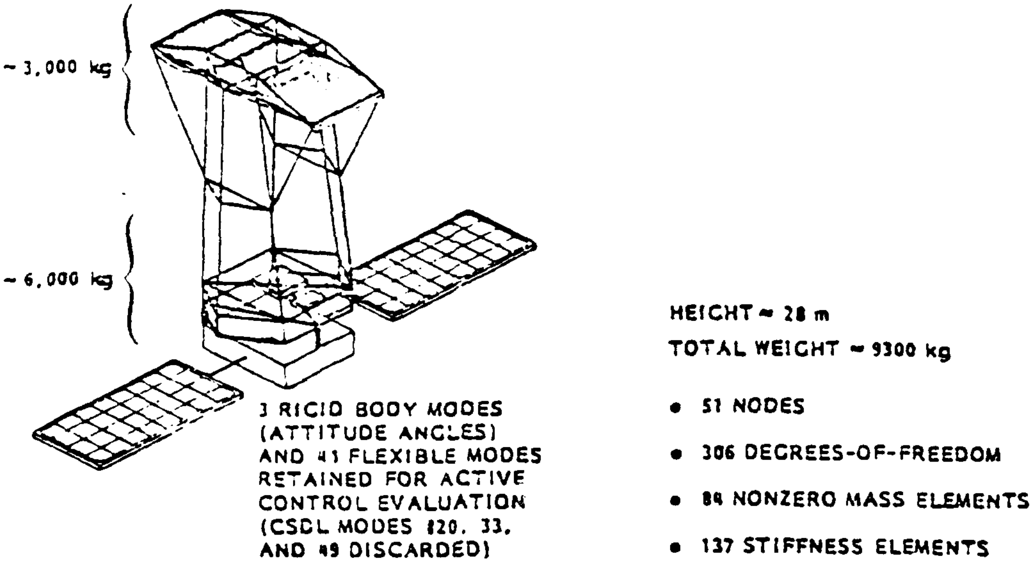

~ 3,000 kg
~ 6,000 kg
3 RIGID BODY MODES (ATTITUDE ANGLES) AND 41 FLEXIBLE MODES RETAINED FOR ACTIVE CONTROL EVALUATION (CSDL MODES #20, 33, AND 49 DISCARDED)
HEIGHT ~ 28 m
TOTAL WEIGHT ~ 9300 kg
• 51 NODES
• 306 DEGREES-OF-FREEDOM
• 84 NONZERO MASS ELEMENTS
• 137 STIFFNESS ELEMENTS

Figure 2: CSDL No. 2 Model (Artist's Conception)

### 2.1.1 Nonlinear Models

Analyses provided by such programs as NASTRAN are quite adequate for small LSS motions, but become suspect as displacement amplitudes increase, particularly with regard to the size of the rotational rigid body modes. Large rigid body angles can usually be accommodated, but if the angular rates grow too much, certain nonlinear dynamic effects have to be modeled, even though structural deformations can still be represented by linear equations. These effects can be modeled by employing Grumman's SATSIM or SPACE14 programs (see Appendix A).

A version of CVA has recently been postulated to identify certain general classes of nonlinearities ( Larimore, 1987), but this more complex version of CVA has yet to be fully validated in code. Similarly, a stochastic version of MPC can feasibly be developed to work with nonlinear models in a manner compatible with CVA identification; software validation is now underway on this code. The focus of the proposed effort thus will remain on linear reduced order models.

Although structural and LSS models of varying degrees of dynamic accuracy will be used to develop and adapt software, the primary baseline model at this time is expected to be the CSDL2 structure, developed for the ACOSS program (Strunce, et al., 1980). See Figure (2). It is expected that sufficient modes can be retained from the baseline model to display meaningful dynamic "pathologies" for proper analysis of the identification and control techniques, and yet result in a model small enough for efficient simulation.

### 2.1.2 Factors in Model Selection

A major reason for pursuing this entire line of research is our feeling that the proposed identification and control methods offer great potential for being more cost effective, and even outperforming, currently available methods. This belief, as stated in earlier sections, is based on the excellent results obtained using CVA for identification in a real-time wing flutter suppression application, and on the preliminary results we have thus far obtained in LSS applications as part of Phase I of this effort. MPC has not been developed to this extent, but all of its applications in a simulation environment point to successful real-time LSS application. These applications include terrain-following (Reid, et al., 1981) and battle damage reconfiguration of aircraft flight control systems (Carroll and Mahmood, 1986).

It is of great use, then, to compare LSSICS results with those of the other schemes, and this comparison can be more meaningful if the same baseline model is used. We are thus led to give serious consideration to using a structure developed by Draper Laboratory for the ACOSS program, the CSDL2 model. This model and its attributes is detailed in Strunce, et al., (1980). This model represents a wide-angle, three-mirror optical space system, about 90 feet high and weighing 10 tons. Through the kindness of Tim Henderson and Dan Hegg of CSDL, we have obtained a data tape of NASTRAN CSDL2 output, and have developed code for reducing these data to linear state space models.

The full model has upward of 50 modes. Our Phase II work will establish an appropriate number of modes for our "truth" model, and for control design. The "truth" model will perhaps consist of 15 to 20 modes, and the control design model at least half of that.

### 2.1.3 Disturbance Modeling

Disturbance effects are not modeled directly in the control synthesis process, but are a part of the "truth" model. Additive white noise or low order Markov processes will be added to the basic dynamic and measurement equations. Random secular, or bias, quantities can similarly be added. These would simulate certain environmental effects such as solar pressure, or some types of system failure - e.g., control jet stuck on.

We note also that CVA identification also obtains models for the above disturbance processes. This information would be of significant use to the FDI and reconfiguration functions (not addressed in this report) of an LSS control system.

Because of the desire to analyze more detailed LSS models, and to perform other types of analyses, we realize the need for greater computational resources than were utilized in Phase I of this project. We will utilize the resources of our subcontractor, Grumman Corporate Research Center, to continue this analysis in Phase II.

## 2.2 CVA Identification

In this section, the technical approach to LSS model identification is discussed in detail including the CVA approach, selection of model order and structure using entropy based methods such as the Akaike Information Criterion (AIC), and computational aspects.

### 2.2.1 Canonical Variate Analysis

The generalized canonical variate analysis approach has recently provided a completely general solution to the static reduced rank stochastic prediction problem which is well defined statistically and computationally even when some or all of the various covariance matrices are singular (Larimore, 1986). All other previous methods in the statistical literature do not address the general problem. For the general time series system identification problem, this result guarantees that the solution to the problem is always well conditioned and produces a statistically meaningful solution. In this section, the background for the CVA method and details of this new result are given.

The analysis of canonical correlations and variates is a method of mathematical statistics developed by Hotelling (1936; also see Anderson, 1958). Concepts of canonical variables for representing random processes were explored by Gelfand and Yaglom (1959), Yaglom (1970), and Kailath (1974). The initial application of the canonical correlation analysis method to stochastic realization theory and system identification was done in the pioneering work of Akaike (1974a, 1975, 1976). This initial work has a number of limitations such as no system inputs, no additive measurement noise, substantial computational burden involving numerous SVD's, a heuristic set of decisions for choosing a basis for representation of the system, and a number of approximations including computation of the AIC criterion for decision on model order.

Some important generalizations and improvements in Akaike's canonical correlation method have recently been made by Larimore (1983b). These include generalization to systems with additive measurement noise and with inputs including feedback controls. A major departure of the approach from previous work is the use of a single SVD to optimally choose $k$ linear combinations of the past for prediction of the future. The very natural measure of quadratically weighted prediction errors at possibly all future time steps is used. Formulated as such a prediction problem, it is shown how a generalized canonical variate analysis gives the solution explicitly. The interpretation of canonical variates as optimal predictors is central in motivating interest in such a problem formulation and is scarcely found in the statistical literature (Larimore, 1986). The optimal k-order predictors are not in general recursively computable, but the optimal state-space structure for approximating them is expressed simply in terms of the canonical variate analysis. The problem of finding an optimal Hankel norm reduced order model (Adamjan et al, 1971; Kung and Lin, 1981) is related to the canonical variate approach (Camuto and Menga, 1982; Larimore, 1983b). The balanced realization method is a particular case of the generalized canonical variate analysis (Desai and Pal, 1984). To more concisely discuss the related research, the problems of identification, reduced order modeling and filtering can be described as follows (Larimore, 1983b).

Consider the problem of choosing an optimal system or model of specified order for use in predicting the future evolution of the process. Consider the past vector $p_t$ consisting of past outputs $y_t$ and inputs $u_t$ before time $t$ and the future vector $f_t$ of outputs at time $t$ or later so

$$p_t^T = (y_{t-1}, y_{1-2}, \cdots, u_{t-1}, u_{1-2}, \cdots)^T, \quad f_t^T = (y_t, y_{t+1}, \cdots)^T \tag{7}$$

We assume that the processes $y_t$ and $u_t$ are jointly stationary and denote the covariance matrices among $f$ and $p$ as $\Sigma_{ff}, \Sigma_{pp}$, and $\Sigma_{fp}$.

The major interest is in determining a specified number $k$ of linear combinations of the past $p_t$ which allow optimal prediction of the future $f_t$. The set of $k$ linear combinations of the past $p_t$ are denoted as a $k \times 1$ vector $m_t$ and are considered as $k$-order memory of the past. The optimal linear prediction $\hat{f}_t$ of the future $f_t$, which is a function of a reduced order memory $m_t$, is measured in terms of the prediction error

$$E\{\| f_t - \hat{f}_t \|_{\Lambda^\dagger}^2\} = E\{(f_t - \hat{f}_t)^T \Lambda^\dagger (f_t - \hat{f}_t)\} \tag{8}$$

where $E$ is the expectation operation and $\Lambda$ is an arbitrary positive semidefinite symmetric matrix so that $\Lambda^\dagger$ is an arbitrary quadratic weighting that is possibly singular. The optimal prediction problem is to determine an optimal $k$-order memory

$$m_t = J_k p_t \tag{9}$$

by choosing the $k$ rows of $J_k$ such that the optimal linear predictor $\hat{f}_t(m_t)$ based on $m_t$ minimizes the prediction error (8).

As derived in Larimore (1986), the solution to this problem in the completely general case where the matrices $\Sigma_{ff}$, $\Sigma_{pp}$, and $\Lambda$ may be singular is given by the generalized singular value decomposition as stated in the following theorem.

**Theorem 1.** Consider the problem of choosing $k$ linear combinations $m_t = J_k p_t$ of $p_t$ for predicting $f_t$, such that (8) is minimized where $\Sigma_{pp}$, and $\Lambda$ are possibly singular positive semidefinite symmetric matrices with ranks $m$ and $n$ respectively. Then the existence and uniqueness of solutions are completely characterized by the $(\Sigma_{pp}, \Lambda)$-generalized singular value decomposition which guarantees the existence of matrices $J$, $L$, and generalized singular values $\gamma_1, \cdots, \gamma_r$ such that

$$J \Sigma_{pp} J^T = I_m, \quad L \Lambda L^T = I_n, \quad J \Sigma_{pf} L^T = Diag(\gamma_1 \geq \cdots \geq \gamma_r \geq 0, \cdots, 0) \tag{10}$$

The solution is given by choosing the rows of $J_k$ as the first $k$ rows of $J$ if the $k$-th singular value satisfies $\gamma_k \geq \gamma_{k+1}$. If there are $r$ repeated singular values equal to $\gamma_k$, then there is an arbitrary selection from among the corresponding singular vectors, i.e. rows of $J$. The minimum value is

$$\min_{rank(J_k \Sigma_{pp} J_k^T) = k} E\{\| f_t - \hat{f}_t \|_{\Lambda^\dagger}^2\} = tr \Lambda^\dagger \Sigma_{ff} - \gamma_1^2 - \cdots - \gamma_k^2 \tag{11}$$

This result not only gives a complete characterization of the solutions in selecting optimal predictors $m_k$ from the past $p_t$ for prediction of the future $f_t$, but the reduction in prediction error for all possible selections of order $k$ is given simply in terms of the generalized singular values. This is of great importance since it avoids having to do a considerable amount of computation to determine what selection of order is appropriate in a given problem.

Different selections of the weighting matrix $\Lambda$ can be used for different purposes. A number of classical reduced rank statistical analysis problems of static variables, i.e. with independence from 'time' to 'time', can be formulated and solved by the generalized CVA of Theorem 1. In the classical canonical correlation analysis problem, $\Lambda = \Sigma_{ff}$. In the principal components analysis problem, the 'past' and 'future' are the same space and in addition $\Lambda = I$. A generalization of this with the 'past' and 'future' different is the principal component analysis of instrumental variables where $\Lambda = I$ (Rao, 1965). The only consideration of the case of singular covariance matrices for the ca.ıonical correlation analysis problem is by Khatri (1976). The solution is considerably more complicated and is not related to a computational procedure such as the SVD. Also it does not address the general CVA problem with $\Lambda$ an arbitrary positive semidefinite matrix. For system identification, the use of the weighting matrix $\Lambda = \Sigma_{ff}$ results in a near maximum likelihood system identification procedure (Larimore, Mahmood and Mehra, 1984).

### 2.2.2 Selection of State Space Order and Model Structure

The generalized CVA method allows the determination of the fit of the various state space models and the selection of the best model state order before computation of the state space models. This "built-in" feature of the CVA method has great significance with regard to the LSS problem, for which it may be quite risky merely tu truncate high frequency modes. The model order selection problem has several aspects:

(1) the selection of the best model order in some statistical sense based upon the observed data, and

(2) model order reduction based upon some criterion such as control performance error, and not merely magnitude of modal frequency.

Both of these issues are addressed by approaching the problem using the generalized canonical variate analysis. In the remainder of this section, the system identification problem is discussed including the statistical order determination and order reduction problems.

Consider the general case of the reduced order filtering and modeling problem: given the past of the related random processes $u_t$ and $y_t$, we wish to model and predict the future of $y_t$ by a $k$-order state $x_t$ and state-space structure of the form

$$x_{t+1} = Fx_t + Gu_t + w_t \tag{12}$$

$$y_t = Cx_t + Au_t + Bw_t + v_t \tag{13}$$

where $x_t$ is the state and $w_t$ and $v_t$ are white noise processes that are independent with covariance matrices $Q$ and $R$ respectively. Equations (12) and (13) may be seen as discretizations of Equations (5) and (6), with allowance here for feedforward dynamics and random processes. The matrices $A$ and $B$ have a different context here than in (5). The white noise processes model the covariance structure of the error in predicting $y_t$ and $x_{t+1}$ from $u_t$ and $x_t$. A special case of the reduced-order filtering problem is the transfer function approximation problem where $u_t$ and $y_t$ are the input and output processes and an approximate state-space model is desired.

In the computational problem given finite data, the past and future of the process are taken to be finite of length $d$ lags so

$$p_t^T = (y_{t-1}^T, \cdots, y_{t-d}^T, u_{t-1}^T, \cdots, u_{t-d}^T)^T, \quad f_t^T = (y_t^T, \cdots, y_{t+d-1}^T)^T \tag{14}$$

Akaike (1976) proposed choosing the number $d$ of lags by least squares autoregressive modeling using recursive least squares algorithms and choosing the number of lags as that minimizing the AIC criterion discussed below. This insures that a sufficient number of lags are used to capture all of the statistically significant behavior in the data. This procedure is easily generalized to include the case with inputs $u$. In the model identification problem, by using the weighting matrix $\Lambda = \Sigma_{ff}$ the identified system is close to the maximum likelihood estimation solution (Larimore, Mahmood, and Mehra, 1984). The generalized SVD of Theorem 1 above determines a transformation $J$ of the past that puts the state in a canonical form so that the memory $m_t = J_k p_t$ contains the states ordered in terms of their importance in modeling the process. The optimal memory for a given order $k$ then corresponds to selection of the first $k$ states.

For the determination of model state order, recent developments in the selection of model order and structure based upon entropy or information will be used. Such methods were originally developed by Akaike (1973) and involve the use of the Akaike Information Criterion (AIC) for deciding the appropriate order of a statistical model. The AIC for each state order $k$ is defined by

$$AIC(k) = -2 \log p(Y^N, U^N; \hat{\theta}_k) + 2M_k \tag{15}$$

where $p$ is the likelihood function, based on the observations $(Y^N, U^N)$ at $N$ time points, with the maximum likelihood parameter estimates $\hat{\theta}_k$ using a $k$-order model with $M_k$ parameters. The model order $k$ is chosen with the minimum value of AIC($k$). A predictive inference justification of the use of an information or entropy based criterion such as AIC is given in Larimore (1983a) based upon the fundamental principles of sufficiency and repeated sampling. The number of parameters $M_k$ in the state space model (12) and (13) is determined by the general state space canonical form as in Candy et al (1979) as

$$M_k = 2kn + km + nm + n(n+1)/2 \tag{16}$$

where $n$ and $m$ are the dimensions of the output and input vectors $y_t$ and $u_t$ respectively. This is far less than the number of elements in the various state space matrices.

In the paper of Akaike (1976), an approximate procedure for evaluating the AIC using the sample canonical correlations is given. This procedure has been found to be highly approximate because the canonical correlation theory upon which it is based assumes independence of variables which is violated for a correlated time series. The exact AIC can be evaluated directly from the generalized CVA described above, although it requires substantial computation. Recently, efficient algorithms for accurate evaluation of the AIC from the generalized CVA have been developed at BTS.

In choosing model order, the risks of introducing bias into the model in choosing too low an order must be weighed against introducing additional variability in choosing too high an order (Larimore and Mehra, 1985). Having selected the optimal statistical order for the model of the observed data, additional order reduction may be appropriate in determining a filter or controller in various applications. As discussed in Larimore (1983b), the general weighting criterion used in the generalized CVA gives a natural procedure for such additional order reduction. This is very important in control design for adaptive control. A general weighting can be used which reflects the important frequencies which are necessary to control.

Once the optimal $k$-order memory $m_t$ is determined, state-space equations of the form (12) and (13) for approximating the process evolution are easily computed by a simple multiple regression procedure (Larimore, 1983b). Since the CVA system identification procedure involves the state space model form, it has the major advantage that the model is globally identifiable so that the method is statistically well-conditioned in contrast to ARMA modeling methods (Gevers and Wertz, 1982). Furthermore, since the computations are primarily a SVD, the computations are numerically stable and accurate with an upper bound on the required computations. Thus the method is completely reliable, and has been demonstrated as such in the adaptive flutter suppression system in wind tunnel tests involving reidentification of the system dynamics tens of thousands of times. From the theory of the CVA method (Larimore, Mahmood and Mehra, 1984), it can be shown that there are no difficulties such as biased estimates caused by the presence of a correlated feedback signal.

### 2.2.3 Computational Aspects

For complex space structures, the dimension of the matrices can easily be in the hundreds or larger. The computation of the SVD for such large matrices can require considerable time on conventional serial computers. The availability of multiprocessors such as the CRAY or the INMOS Transputer makes possible considerable reduction in the required computation time. We propose here to work with models sized for the Micro-VAX.

A very efficient algorithm for computing the usual SVD has been recently devised for highly parallel systolic arrays by Brent and Luk (1985). This algorithm was recently generalized (Luk, 1985) in one particular way to compute the B-SVD (Van Loan, 1976) which is different from the generalized SVD required for the CVA method. An $n \times n$ square systolic array of processors requires communication only between the nearest neighbor processors in synchrony with the computational cycle of all the processors. The computation of the SVD of a $n \times n$ matrix using a $n \times n$ array of processors requires only order $n$ processor cycles as compared to order $n$ cubed for a serial computer with a single processor. Such parallel processors and algorithms could make routine the analysis of very large sets of variables such as arise naturally in the canonical variate analysis of large order or nonlinear systems.

Modern computer algorithms (Golub, 1969) for canonical correlation analysis use a standard singular value decomposition to compute the generalized singular value decomposition (see Equation (10)) with $\Lambda = \Sigma_{ff}$ by first finding square root factors of $\Sigma_{pp}$ and $\Lambda$, and then doing a standard singular value decomposition on $A = \Sigma_{pp}^{-1/2}\Sigma_{pf}(\Lambda^{-1/2}) = QSR^T$ where $QQ^T = I = RR^T$ and $S$ is diagonal. Then the generalized singular value decomposition is given by $J = Q^T\Sigma_{pp}^{-1/2}, L = R^T\Lambda^{-1/2}$ and $D = S$. Thus the joint orthonormalization of $p_t$ and $f_t$ in the norms $\Sigma_{pp}$ and $\Lambda$ to give the canonical covariance structure $D$ is very naturally viewed as a generalized singular value decomposition both in terms of the simple reduction discussed in Theorem 1 as well as the actual computational algorithms. This can be determined computationally using a standard singular value decomposition which is numerically very accurate and stable as compared with the earlier eigenvalue computational methods (Bjorck and Golub, 1973). An open topic is the investigation of numerical methods that directly compute the particular generalized SVD rather than transforming the problem to the standard SVD. Such a direct approach may have better overall numerical accuracy.

A second problem is specified in terms of the observed data given as N observations $(p_1, \ldots, p_N) = B$ and $(f_1, \ldots, f_N) = C$ on $p$ and $f$ respectively. The usual sample covariances are computed as $\Sigma_{pp} = BB^T, \Sigma_{pf} = BC^T$ and $\Sigma_{ff} = CC^T$ which mathematically are used in the generalized singular value decomposition. Numerically, however, the formation of these products defining the sample covariances results in a halving of the numerical precision of the computation. In the case of given data, Bjorck and Golub (1973) give computational procedures that avoid these squaring operations and operate directly on the observed data.

## 2.3 Model Predictive Control

A critical need for a robust, adaptive controller is a reliable system model of appropriate order. We assume that an appropriate identification method exists to supply these models to the controller. Reduced order model predictive control gains can be designed adaptively in real time as the models are updated. Control synthesis can thus take place by rote (pre-stored algorithms), without intensive engineering analysis, as the system model changes. A form of model predictive control which involves the use of the Singular Value Decomposition (SVD) has been outlined by Reid et al. (1981), and applied to reconfigurable flight control systems by Carroll et al., (1986). Because SVD and MPC are similar computationally, there is an opportunity for a more efficient algorithmic structure, especially if systolic array processors are utilized. This particular type of controller is thus very compatible with computationally stable identification methods (Section 2.2) and is now briefly described.

The Model Predictive Control (MPC) technique has been developed as the next generation of a widely used control technique known as Model Algorithmic Control (MAC). There has lately been a growing interest in the class of MAC-type controllers, known also as "predictive controllers", and MAC is probably the earliest one in this class reported in the literature, as noted above.

MPC has its origins in adaptive controllers developed for industrial processes. There is an input/output structure to this type of controller which is amenable to adaptation to system parameter changes, without the need to understand explicitly the dynamics behind the particular change. This input/output structure eliminates the formal need for full state feedback required by most other multivariable control synthesis methods, and allows for greater structural compatibility with robust identification methods. Another serious limitation of the earlier controllers is that they typically use an impulse response model of the plant to be controlled (Richalet, et al. 1978) and therefore cannot be used for an open-loop unstable plant.

The MAC and MPC techniques are based on the same principles. These techniques are fundamentally and philosophically different from "feedback" controllers such as the LQ regulator and its variants, in which there is an explicit notion of "feedback" of the current state to derive a closed-loop control law. Instead, closed loop control is achieved by accomplishing, at every cycle of the digital control loop, the following functions: (a) *predict* the future zero input response of the plant using an observer or a model of the plant over a selected "horizon of prediction". The duration of this prediction interval is a key stability and robustness parameter. The prediction must be closed loop, i.e., based on available measurements of the actual output response. A Kalman filter or observer can provide an estimate of the current state, based on measured outputs and the system mathematical model. (b) Compute a *desired future reference trajectory* along which the system is desired to be moved over the "horizon of prediction" to a set point (which can be zero for a regulator problem, or a function of time for tracking, e.g., terrain following or on-orbit target tracking). (c) Find a *control sequence* to minimize the Euclidean distance between the predicted output and the desired trajectory. Reid, et al., (1981) have developed a way to generate the future control sequence by formulating the error criteria as a linear least squares problem. This is done by adding "output smoothing points" - i.e., by sampling outputs at a faster rate than that at which the control is updated. There is thus an *implicit* control of the system state even though there is no explicit state control. The result is that the output trajectory remains close to the reference trajectory even between control update times; this has an effect of reducing system internal energy, which for the LSS is directly related to large excursions of the state variables (displacement or modal coordinates) from their reference level (Colson, 1978).

The difference between MAC and MPC lies in the fact that MAC uses an impulse response model of the plant to predict its future output and hence cannot be used to control an open-loop unstable plant. To overcome this problem, a special formulation which uses a state-space model of the plant, MPC, has been developed. However, the main advantage of MPC/MAC is that there is a clear and transparent relationship between system performance and various parameters embedded in the design procedure. This feature is particularly useful in LSS applications, where effective reduced order modeling is vital. SVD identification clearly enhances this feature as well. It should finally be pointed out that the structure of MPC allows for decentralized, or distributed control designs. The peculiarities of LSS dynamics often dictate such a design (Kosut and Lyons, 1984).

In this section we develop an exact model of the algorithm for control computation, find an explicit form of the control law, and place MPC in the standard control design framework so that stability, gain and phase margin, robustness, etc., can be ascertained *a priori*. The designer can then iterate on various design parameters to meet the specifications of the particular LSS system.

There are many variations of predictive controllers such as DMC (Cutler and Ramaker, 1979/80), IMC (Garcia and Moyari, 1982), etc., but here we closely follow the developments in Reid et al. (1981) and Carroll and Mahmood, (1986).

### 2.3.1 Basic Elements of MPC

There are five elements in MPC:

(i) A dynamic plant to be controlled:

$$\begin{aligned} x(k+1) &= Fx(k)+Gu(k), \quad x(0)=x_0 && (17)\\ y(k) &= Cx(k) && (18) \end{aligned}$$

We assume that $x(k), u(k)$ and $y(k)$ are of dimension n, m and p, respectively. Also $F$ and $G$ are the discrete system and control matrices, respectively. A feedforward term is straightforward to add in Equation (18), but is avoided here for clarity. It is not vital that the plant model be linear.

(ii) An internal model of the plant having the same input-output dimension $m \times p$ as that of the best representation of the actual plant.

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{F}\tilde{x}(k)+\tilde{G}u(k), \quad \tilde{x}(0)=\tilde{x}_0 && (19)\\ \tilde{y}(k) &= \tilde{C}\tilde{x}(k) && (20) \end{aligned}$$

(iii) A p-dimensional reference trajectory $y_r(k)$, along which the system is desired to be guided to a set point. There are many variations of generating this desired trajectory. In our approach, $y_r(k)$ is a preferably smooth curve initialized on the current output of the actual plant $y(k)$ that leads $y(k)$ to a possibly time varying p-dimensional set point $s(k)$. In this analysis $y_r(k)$ evolves on:

$$\begin{aligned} y_r(k+1) &= \Lambda_\alpha y_r(k)+(I-\Lambda_\alpha)s(k), \quad y_r(k)=y(k); && (21)\\ |\lambda_i(\Lambda_\alpha)| &< 1, \ for\ all\ i, \end{aligned}$$

where $\lambda_i(X)$ is the $ith$ eigenvalue of X.

Usually $\Lambda_\alpha$ is a diagonal matrix, i.e., $\Lambda_\alpha = diag(\alpha_i)$ and $0 < \alpha_i < 1$. The reference trajectory in Equation (21) has first order dynamics but may not; splines or some other mechanism of generating the reference trajectory can be used. However, it is easy to see that the solution of (21) is, if $s(k) = s$,

$$y_r(k+N) = \Lambda_\alpha^N y(k) + (I - \Lambda_\alpha^N)s \qquad (22)$$

The $\alpha_i$ thus are like discrete first order time constants which can be adjusted for a certain accuracy in $y_r$ after $N$ steps. Any method chosen should be simple, as the calculation is done every control update.

(iv) A closed-loop prediction scheme for predicting the future output of the plant according to

$$y_p(k+1) = \tilde{y}(k+1) + y(k) - \tilde{y}(k) \qquad (23)$$

The $y_p(k+1)$ may be considered to be elements of $Y_t^f$.

Again, we have used this scheme for the sake of simplicity but the user can supply his own routine for generating the predictions. An acceptable implementation is to use the zero input response to the state at $k, x(k) : CF^i x(k), \quad i = 1, ..., N$, for each of the $N$ outputs over the horizon of prediction.

(v) A quadratic cost functional J based on the error between $y_p(k)$ and $y_r(k)$ over a finite horizon N:

$$J = tr\sum_{i=1}^{N}[Q(k+i)e(k+i)e^T(k+i) + R(k+i-1)u(k+i-1)u^T(k+i-1)] \tag{24}$$

where $Q(\cdot)$ and $R(\cdot)$ are positive semi-definite possibly time-varying weights and typically $e(k+i)$ is the error to be minimized in the overdetermined problem (see below). In most applications $R(\cdot)$ is set to zero. This is acceptable since physical limits on the control element excursions are easily incorporated into the logic for deriving the control sequence.

Given (i) - (v), MPC finds an optimal control sequence $u^*(k+i-1),\ i = 1,2,...,N$ by minimizing $J$ over the admissible input sequence $u(k+i-1) \epsilon\ \Omega,\ i = 1,2,...,N$. Once this sequence is computed, the first element, i.e., $u^*(k)$, is applied to the actual plant and the entire process is repeated all over again. Note that the input/output nature of MPC, and specifically the direct incorporation of a predicted desired output in the calculation of the optimal control sequence, mean that merely a simple structuring of the output setpoints $s(k)$ will cause MPC to behave as a regulator (shape control) or a tracker (vibration control).

Usually a control law is designed on the assumption that the actual plant is the same as the nominal model although the latter is almost invariably different from the former. The robustness of the designed control law is evaluated on the nominal model which specifies the stability neighborhood around it. If the actual plant lies in this region the nominal control law will guarantee the closed-loop stability for the actual plant.

As Equation (24) implies, MPC control can be formulated as a weighted least squares problem, which develops as follows[2]:

Future output $y(k)$ is related to present state $x(0)$ and future inputs $\{u(0), u(1), ..., u(k-1)\}$ via the discrete equation

$$y(k) \ = \ y_{zi}(k) \ + \ y_{zs}(k) \tag{25}$$

(Note that current time is referenced to $k = 0$.) In Equation (25), $y_{zi}$ is the zero input response,

$$y_{zi}(k) \ = \ \tilde{C}\tilde{F}^k x(0),$$

and $y_{zs}$ is the zero state response, or (see, e.g., Reid 1983),

$$y_{zs}(k) \ = \ \sum_{i=0}^{k-i} h_d(k-i)u(i) \tag{26}$$

This formulation assumes that the control input is piecewise constant over the control sample interval, $T_c$, and indexed so that

$$u(t) \ \equiv \ u(k),\ t\ \epsilon(kT_c, (k+1)T_c)$$

[2] This development is based on Reid, et al., (1981), but is expanded to handle MIMO problems.

We now establish other preliminaries: let NSM be the number of "output smoothing" points per control update. We assume throughout this rep rt that the system discretization and output sampling periods, $T_d$, are the same. This assumption is one of convenience which may be relaxed at a future time without much difficulty, if warranted. Thus, $T_c = \text{NSM} \cdot T_d$. If $L$ is the number of discrete controls to be predicted into the future, then the total number of samples in the prediction horizon is NSM $\cdot$ $L$. It turns out that NSM and $L$, in addition to the cost weighting matrices $Q$ and $R$, are key MPC design parameters.

Equation (26) represents a discrete convolution process, with $h_d(k)$ being the discrete impulse response function

$$h_d(k) = \tilde{C}\tilde{F}^{k-1}\tilde{G} \tag{27}$$

The MPC problem then involves solving Equation (24) for the sequence $\{u(k)\}$, taken over the horizon of prediction, with $y(k)$ in (24) replaced by the predicted desired output, $y_d(k)$. This is done by considering the NSM $\cdot$ $L$ outputs in the horizon of prediction simultaneously, substituting Equation (26) into (25), and formulating as a vector-matrix equation

$$Y_{zi} + \bar{H}U = Y_d \tag{28}$$

We note here that the $Y$-terms in Equation (28) consist of a string of NSM $\cdot$ $L$ $p$-dimensional output vectors:

$$Y_d = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(\text{NSM} \cdot L) \end{bmatrix}, \quad Y_{zi} = \begin{bmatrix} y_{zi}(1) \\ y_{zi}(2) \\ \vdots \\ y_{zi}(\text{NSM} \cdot L) \end{bmatrix}$$

The $y_d(k)$ are generated according to designer option, performance requirements, etc. (see, for example, Equation (21)). This is usually done according to whether the control objective is to achieve some constant value ("setpoint") $s$, the regulator case, or to follow a varying reference path, the tracking case. The LSS problem has examples of both general cases: regulation of shape (eg., antenna) and step changes in attitude are examples of a regulator mode, while target tracking is obviously a tracker mode example.

The explicit formulation of $\bar{H}U$ depends on such issues as the desirability of using impulse response or step response functions, etc. Since these variants are mathematically equivalent, the practical issues are then computational accuracy, availability, storage and run-time efficiency - tradeoffs which we propose to analyze in this project. A favored mechanization is to eliminate some summations by using step response functions instead of impulse response functions:

$$r(k) = \sum_{i=1}^{k} h_d(i) = r(k-1) + h_d(k)$$

where $r(k)$ is the step response at sample point $k$ to a unit step input applied at reference sample point $k = 0$. Rewriting Equation (26) thus means that

$$U \triangleq \begin{bmatrix} u(0) \\ \Delta u(1) \\ \vdots \\ \Delta u(L-1) \end{bmatrix}, \quad \bar{H} = \begin{bmatrix} \mathbf{h}_1 & 0 & \cdots & 0 \\ \mathbf{h}_2 & \mathbf{h}_1 & \cdots & 0 \\ \vdots & & & \vdots \\ \mathbf{h}_L & \mathbf{h}_{L-1} & \cdots & \mathbf{h}_1 \end{bmatrix} \tag{29}$$

where the $\Delta u(i)$ are changes to the previous applied control, i.e.,

$$u(k) = u(0) + \Delta u(1) + \ldots + \Delta u(k) = u(k-1) + \Delta u(k)$$

$U$ is an $mL \times 1$ vector array. $\bar{H}$ is very similar in form (and information content) to the Hankel matrix. The $\mathbf{h}_k$ in $\bar{H}$ are submatrices consisting themselves of NSM transfer function matrices each of size $p \times m$. Dimension $(\mathbf{h}_i) = p \cdot \mathrm{NSM} \times m$ in the following:

$$\mathbf{h}_k = \begin{bmatrix} \sum_{i=1}^{\mathrm{NSM}(k-1)+1} h_d(i) \\ \vdots \\ \sum_{i=1}^{\mathrm{NSM}\cdot k} h_d(i) \end{bmatrix} = \begin{bmatrix} r(\mathrm{NSM}(k-1)+1) \\ \vdots \\ r(\mathrm{NSM}\cdot k) \end{bmatrix} \tag{30}$$

where the $h_d$ are defined in Equation (27). Equation (30) is a corrected version of Equation (15) in Reid, et al., (1981), and it is extended to the MIMO case, as are the other equations in this development.

Having formulated the MPC problem, the solution is to solve Equation (28) for $U$. Reid's contribution was to simultaneously control internal plant energy and convert Equation (28) into an overdetermined linear equation problem by adding the NSM output smoothing points per control cycle (Reid, et al., 1981). The usual least squares "approximate" solution methods then apply. The weighted criterion is needed to keep (28) as valid as possible, and also (an extension of Reid, developed by him and Carroll and Mahmood (1986)) to penalize large control excursions. The cost criterion (24) then becomes

$$J = [\bar{H}U - (Y_d - Y_{zi})]^T Q [\bar{H}U - (Y_d - Y_{zi})] + U^T R U \tag{31}$$

which has for a solution

$$U^* = (\bar{H}^T Q \bar{H} + R)^{-1} \bar{H}^T Q (Y_d - Y_{zi}) \tag{32}$$

Note: (1) Equation (32) solves simultaneously for the $L$ future controls (cf., Equation (29)), but only the first element, $u(0)$, is actually applied to the system; in a closed loop operation the problem will be reformulated and solved at each control sample time. While much of Equation (28) thus seems to be "wasted", it is necessary to point out that the "horizon of prediction" $L$ has a very strong effect on closed loop *performance*.

(2) The arrays defining $U^*$ can rapidly become highly unwieldy. This is especially true in most LSS applications. However, two mildly restrictive assumptions can be exploited to *significantly* reduce required core:

(i) keep NSM fixed in value for each of the $L$ future control settings;

(ii) Let

$$Q = diag(Q_1, ..., Q_L)$$

where $Q_i$ is a $p$NSM × $p$NSM diagonal submatrix of $Q$.

Assumption (i) was discussed earlier, and probably has no limiting effect of any significance. Assumption (ii) is potentially restrictive, but is very common in practical design algorithms. If these assumptions are allowed, the much lower dimension, symmetric $mL \times mL$ matrix $(\bar{H}^T Q \bar{H})$ may be created directly from

$$[(\bar{H}^T Q \bar{H})_{i,j}] = \left[ \sum_{k=max(i,j)}^{L} \mathbf{h}_{k-i+1}^T \, Q_k \mathbf{h}_{k-j+1} \right]$$

where $\mathbf{h}_i$ are defined in Equation (30). This can be done offline, or whenever the system model is updated. The following computation must be done every update cycle, as the greatest core savings require combining operations directly with the latest outputs (although such an inefficiency in core could be tolerated if processing speed is a greater concern): the $mL \times 1$ vector $\bar{H}^T Q(Y_d - Y_{zi})$ may be created from the $L$ elements of size $m \times 1$

$$\sum_{k=1}^{i} \mathbf{h}_k^T Q_{k+L-i}(Y_d - Y_{zi})_{k+L-i} \tag{33}$$

where $(Y_d - Y_{zi})_k$ are the $pL \times 1$ partitions of the $p \cdot$NSM$\cdot L$-dimension vector $(Y_d - Y_{zi})$. Equation (33) is a corrected and dimensionally upgraded (SISO to MIMO) version of Equation (20) in Reid, et al., (1981). Thus only the $L$ $\mathbf{h}_i$ in Equation (30) are needed to generate $U^*$, not all of $\bar{H}$.

Alternate solution methods to investigate are solving Equation (31) via the Singular Value Decomposition (if memory is not a problem), or solving Equation (28) directly by such a process, as long as certain relationships and/or limits are defined for control values.

### 2.3.2 Closed Loop Stability Analysis

The optimal control sequence $u^*(k)$ can be conveniently written as

$$u^*(k) = L_o y(k) - L_s x(k) + L_c s \tag{34}$$

where

$$\begin{aligned} L_o &= \sum_{i=1}^{N_p} L_i \Lambda_\alpha^i \\ L_s &= \sum_{i=1}^{N_p} L_i C F^i \\ L_c &= \sum_{i=1}^{N_p} L_i (I - \Lambda_\alpha^i), \end{aligned}$$

$N_p$ is the number of output prediction points, and the $L_i$ are the NSM$\cdot L$ row submatrices of $L_G$,

$$L_G \triangleq (L_1, L_2, ..., L_{\mathrm{NSM}\cdot L})$$

The $L_i$ are of size $m \times m$, and $L_G$ is itself defined from Equation (32):

$$\begin{aligned} u^*(k) &= [I_m, 0, ..., 0]U^* \triangleq I_u U^* \\ &= I_u(\bar{H}^T Q \bar{H} + R)^{-1} \bar{H}^T Q (Y_d - Y_{zi}) \\ &\triangleq L_G(Y_D - Y_{zi}) \end{aligned} \tag{35}$$

where $I_m$ is the $m \times n$ identity matrix. Using output relations defined earlier, the result (35) leads directly to Equation (34), for a constant setpoint, $s(k) = s$, and for

$$y_d(k) = (I_p - \Lambda_\alpha^k)s + \Lambda_\alpha^k y(0),$$

a specific version of Equation (21).

Due to the form of Equation (34) we can interpret $L_o$ as the "output feedback" gain matrix, $L_s$ as the "state feedback" gain matrix, and $L_c$ as the "set point feedback" gain matrix. Equation (34) also demonstrates that the MPC-optimal control law is implicitly equivalent to a combination of a state-feedback and an output-feedback control law, and the optimal control loop is equivalent to the set-up in Figure 3. We note, however, that the sequence $u^*(k)$ is best implemented via the approach developed in the preceding subsection, which avoids the explicit need for full state feedback.

It is now straightforward to compute the equivalent closed loop matrix from the block diagram in Figure 3. Clearly, the closed loop states will evolve as

$$\begin{aligned} x_{c\ell}(k+1) &= F x_{c\ell}(k) + G u^*(k) \\ &= F x_{c\ell}(k) + G[L_o y_{c\ell}(k) - L_s x_{c\ell}(k) + L_c s] \\ &= F_{c\ell} x_{c\ell}(k) + G_{c\ell} s \end{aligned} \tag{36}$$

where

$$\begin{aligned} F_{c\ell} &= F + \sum_{i=1}^{N_p} G L_i (\Lambda_\alpha^i C - C F^i), \\ G_{c\ell} &= \sum_{i=1}^{N_p} G L_i (I - \Lambda_\alpha^i), \end{aligned}$$

and $s$ is the command set point.

Again we can interpret $F_{c\ell}$ as the equivalent closed loop matrix and $G_{c\ell}$ as the equivalent input matrix. The optimal closed loop input sequence $u^*(k)$ and output sequence $y(k)$ are related to $x_{c\ell}(k)$ as

$$u^*(k) = (L_o C - L_s) x_{c\ell}(k) + L_c s \tag{37}$$

$$y(k) = C x_{c\ell}(k) \tag{38}$$

Summarizing, the MPC-optimal control sequence $u^*(k)$ can be generated in two ways:

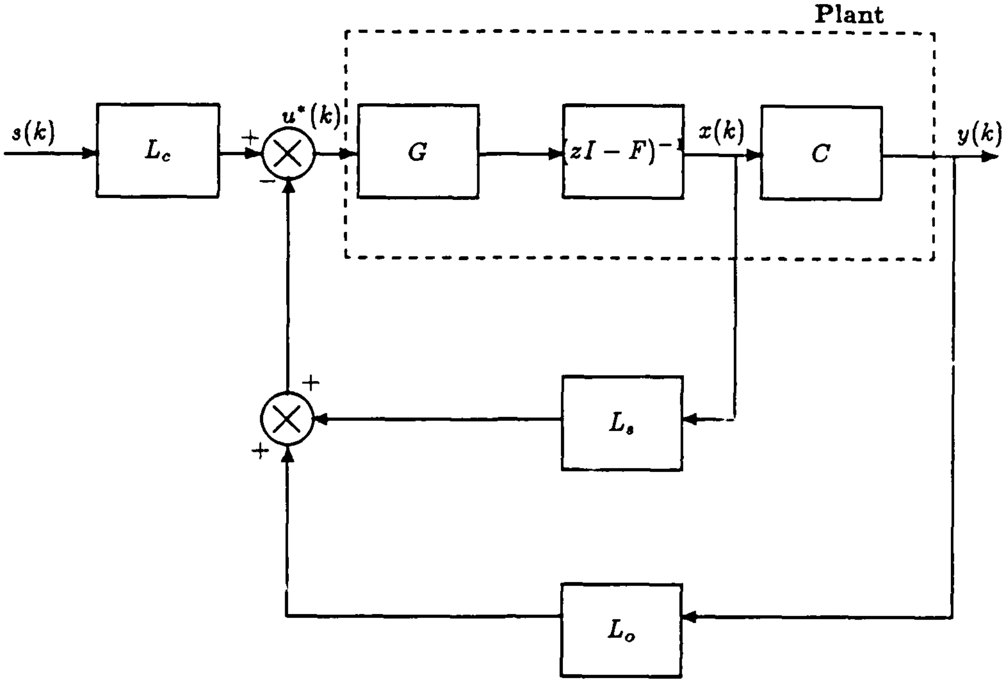


Figure 3: An MPC-Equivalent Loop.

(a) at each k, compute the vector $Z(k) = (Y_d - Y_{zi})$ and then $u^*(k) = L_G Z(k)$; or

(b) compute the matrices $L_o$, $L_s$ and $L_c$ or obtain the optimal control sequence from the setup in Figure 3 or simulate the system derived in Equations (36) to (38).

Method (a) is preferred for on-board implementation: it can run at roughly the same time as method (b), but is more core-efficient. Method (a) can be made fully core efficient, resulting in very substantial core savings, but at some cost in processing speed. This tradeoff awaits more detailed analysis.

It has been shown that the MPC control technique is equivalent to a combination of a constant gain state and output feedback control law and therefore can be analyzed in a classical control framework. The stability, robustness and sensitivity of the MPC technique can be ascertained *a priori* in this technique and the designer can iterate on various design parameters to improve the performance of the loop.

The model predictive control approach provides a transparent relationship between the system performance and design parameters. The inherent robustness of MPC allows it to change automatically some of its control parameters on-line when there is performance degradation. MPC will efficiently generate stabilizing control commands for plants which are open-loop unstable and/or nonminimum phase. The latter is often a characteristic of LSS structures with failed or degraded components. The ability of MPC to adapt optimally in response to system or environment changes brings to the LSS a very intelligent form of active control. It is also clear that passive or "stand-alone" control is achieved both by the feedback nature and robustness of MPC.

## 2.4 Simulation

Simulation analysis was performed using linear discretized "truth" and control design models. Section 3 details the results obtained in Phase I of this project. Early results were obtained using a laboratory-scale flexible model (Tahk and Speyer, 1987), discretized to a basic sample period of 0.05 seconds. This model is dynamically equivalent to the disk-torsion bar problem considered by Cannon and Rosenthal (1984).

PC-MATLAB was the environment for developing a "test" version of MPC. While greatly restricted by size and run-time limitations, this has been an excellent environment for validating extensions of the algorithm, general debugging, and analyzing dynamic behavior. The CVA identification code was not converted to PC-MATLAB because (i) mature VAX/VMS versions exist, and there is little structural upgrading of the CVA algorithm needed for it to be applied to LSS identification; and, (ii) the algorithm is sufficiently detailed that too much Phase I effort would be needed to convert to PC-MATLAB.

A VMS Fortran version of MPC also is being used on larger models. It utilizes features which provide speed at the expense of core - for example, NSM (see Section 2.3) is replaced by an $L$- dimension vector $nsm$, thereby allowing for a variable number of smoothing points per control update over the "horizon of prediction."

# 3 ANALYSIS OF RESULTS

The scope and resources of Phase I of this project preclude an in-depth investigation of the full set of control design issues. We have nevertheless been able to initiate at least a preliminary examination of most of these. The results presented here indicate clearly that the identification and control methods offer feasibility for useful application to flexible structures, but also that more detailed analysis on more realistic models is needed in order to define properly the role for LSSICS. We propose that this analysis be expanded in Phase II of this project.

## 3.1 Algorithm Modifications

Section 2 describes new results in the area of refining and upgrading the Model Predictive Control algorithm. The major features of this effort are:

(i) a fully MIMO version of MPC has been developed and successfully tested

(ii) key errors in the SISO algorithm of Reid, et al., (1981) have been eliminated in the upgraded version

(iii) options have been developed for the following:

- core vs. run time, relating to exploiting certain mildly restrictive assumptions about the design parameters NSM and $Q$ to greatly reduce the core required to generate the control gain matrix (cf, Equation (32), Section 2.3)
- formulating elements of $\bar{H}$ either in terms of impulse response or step response functions
- mechanization based on input/output relations, or upon closed loop stability matrices (see (iv))

(iv) for certain formulations of output quantities, analytic expressions for closed loop MPC have been developed which allow for standard MIMO stability and performance analysis (See Section 2.3). It is thus now possible to observe directly the effect of changes in MPC design parameters, and also in the control design model. The stability analysis segment of MPC is meant to be performed offline (though it could become an online part of accepting or changing design parameters following certain types of model updates). Thus, it is not as core efficient as the input/output version, but does produce the same response results.

It is also very likely that closer examination of the particular formulations (see, for example, Equations (34) and (35)) will indicate a way to develop core-efficient relations of the stability analysis code.

The last point in item (iv) above also applies to the core/run-time tradeoff concerning whether NSM is constant or not. Core savings and algorithmic simplifications result when NSM is constant; however, there is definitely a way to save most of the core now being "wasted" for the variable NSM version of MPC. To exploit this savings would greatly add to the matrix index bookkeeping (arrays would have to be added to the code just for this purpose) and, as stated earlier, the benefits to Phase I of such an undertaking are at best unclear.

Some of the core/run-time issues are quantified in the following analysis summarized in Table 1. In the example, we have used NSM $= 4$, $L = 10$, and $m = p = 5$. Distinct storage is needed for all quantities identified in the table; also, most "core-efficient" quantities are submatrices of the "run-time-efficient" matrixes. Using the above numbers, representing a rather modest LSS design case, we observe a clear difference. The core-efficient code requires 13% of the storage needed by the other formulation, although we note again that there is a way to bring the latter much closer, to perhaps about twice as much required storage. At this level, run-time comparisons would be quite meaningful.

Table 1: MPC Software Implementation

| Issue | Storage | Run-Time | Degree |
|---|---|---|---|
| $U^T = [u_o^T, \Delta u_1^T, ...]$<br>*vs.*<br>$U^T = [u_o^T, u_1^T, ...]$ | Roughly equal | $U^T = [u_o^T, \Delta u_1^T, ...]$ | Minor |
| Constant<br>*vs.*<br>Varying NSM | Constant NSM | Varying NSM | Large, either way |
| $\bar{H}$ using step response<br>*vs.*<br>Impulse/response | Step response | Roughly equal | Minor |
| $T_d$ Value | No effect | Inversely Related | Moderate |
| $L$ Value | Directly Related | Directly Related | Moderate |

At this point, run-time *vs.* storage tradeoffs can be made. Table 1 compares storage data for MPC, and similar data exists for CVA. More or less core will be required depending on the following issues summarized in Table 2. Specific test scenarios must be run to quantify the classifications cited in the table. As stated earlier, higher-level language CVA code is by itself rather completely optimized. Note that it will be possible to do much to offset the excessive storage advantage held by the fixed-NSM option, if sophisticated matrix indexing logic is added. To see that this is so, observe that $\bar{H}$ has $L$ submatrix rows, to contain the "constant control submatrices," **h**. With varying NSM for each of the $L$ future control settings, the $L$ submatrix rows of $\bar{H}$ each have a different "height." Thus, there are $L(L-1)$ distinct **h** matrices (see also Equation

Table 2: Storage Comparison of Different MPC Implementations

| Key Elements of MPC Gain Matrix, Control Computation | Run-Time Efficient (no Assumptions on NSM, $Q$) | | Core-Efficient: NSM = constant, $Q$ diagonal | |
|---|---|---|---|---|
| | General | Example | General | Example |
| $L_G = (\bar{H}^T Q \bar{H} + R)^{-1}\ \bar{H}^T Q$ | $(L \cdot m) \times (\text{NSM} \cdot L \cdot p)$ | 10,000 | $(L \cdot m)(L \cdot m + 1)$ | 2,550 |
| $Y_d, Y_{zi}$ | $2[(\text{NSM} \cdot L \cdot p) \times 1]$ | 400 | $2[(\text{NSM} \cdot L \cdot p) \times 1]$ | 400 |
| $\bar{H}$ | $(\text{NSM} \cdot L \cdot p) \times (L \cdot m)$ | 10,000 | $L \times (\text{NSM} \cdot p \cdot m)$ | 1,000 |
| $Q$ | $(\text{NSM} \cdot L \cdot p)^2$ | 40,000 | $L \times (\text{NSM} \cdot p) \times \text{NSM} \cdot p)$ | 4,000 |
| $R$ | $(L \cdot m) \times (L \times m)$ | 2,500 | $L \times m^2$ | 250 |
| TOTALS | | 62,900 | | 8,200 |

(29)):

$$\bar{H} = \begin{bmatrix} \mathbf{h}_1^{(1)} & 0 & \cdots & 0 \\ \mathbf{h}_2^{(1)} & \mathbf{h}_1^{(2)} & & \vdots \\ \vdots & \vdots & & 0 \\ \mathbf{h}_L^{(1)} & \mathbf{h}_{L-1}^{(2)} & \cdots & \mathbf{h}_1^{(L)} \end{bmatrix} \tag{39}$$

Each $\mathbf{h}_k^{(j)}$ has NSM($k$) $m \times p$ response matrices (cf, Equation (30)), for $1 \leq k \leq L$. Much core can be saved by noting that, for given $k$ and $nsm \triangleq \min_k(\text{NSM}(k))$, the first $nsm$ submatrices in each $\mathbf{h}_k^{(j)}$ are equal, for all j: that is $\mathbf{h}_k^{(1)} = \mathbf{h}_k^{(2)} = \cdots$, through their first $nsm$ submatrices. The issue to be quantified here is whether the payoff in performance merits such an effort.

All issues of this nature will be analyzed and documented further in Phase II in detail sufficient to allow firm decisions based on operational priorities.

## 3.2 CVA Identification Results

To demonstrate the above methods, a system dynamically equivalent to the disk-torsion bar problem considered by Cannon and Rosenthal (1984) was used initially for simulation. This system has a number of similarities with large space structures but is simple in form for simulation and analysis. The system consists of 4 unit weight masses, or disks, arranged along a line and connected by torsion bars modeled by spring and damper elements. The system input is a torsion on mass 2 and the output is the measured rotation displacement of mass 2. All of the disks are free to move about their common axis (Figure 4). The system is an 8 state system with modes at 0.0 for the rigid body mode and at 1.4721, 1.9219, and 2.1598.

For demonstrating CVA, a white noise of standard deviation of 10 was used as the system input, and the output was measured in the presence of a white measurement noise with various standard deviations. The sample rate was 0.01 seconds over 10 seconds for a total of 1000 measurements. The CVA was done using 40 lags.

The results of the system identification are illustrated in Figures 5 and 6. The transfer function frequency response is shown for the true system and for the identified system for the cases of white measurement noises of standard deviations 0.0, 0.0005, 0.005, and 0.05 respectively. The system identification accuracy depends upon the magnitude of the measurement noise. Note that the identification is excellent for the lower measurement noise. As the noise increases, the ability to resolve the lower frequencies diminishes. A plot with linear frequency would show that the lower frequency peaks are progressively narrower and thus more difficult to identify in the presence of white noise. The frequency response phase shows accuracy comparable to the magnitude with high accuracy for low measurement noise.

Table 3: AIC Behavior with State Order

| State | Number of | Measurement Noise Standard Deviation | | |
|---|---|---|---|---|
| Order | Parameters | 0.005 | 0.0005 | 0.0001 |
| 0 | 1 | 2504.84 | 2504.83 | 2504.83 |
| 1 | 4 | -4990.22 | -7498.35 | -7219.62 |
| 2 | 7 | -5125.18 | -7802.97 | -7631.33 |
| 3 | 10 | -5119.79 | -8511.80 | -9016.18 |
| 4 | 13 | -5271.53 | -8514.71 | -9034.42 |
| 5 | 16 | -5399.03 | -8830.58 | -9259.73 |
| 6 | 19 | -6226.97 | -8827.75 | -9263.09 |
| 7 | 22 | -6221.00 | -9074.95 | -9487.54 |
| 8 | 25 | -6270.39 | -9601.93 | -10471.00 |
| 9 | 28 | -6265.29 | -9596.16 | -10466.10 |
| 10 | 31 | -6259.64 | -9590.67 | -10461.30 |
| 11 | 34 | -6255.55 | -9586.03 | -10455.60 |
| 12 | 37 | -6249.71 | -9584.38 | -10449.70 |
| 13 | 40 | -6243.97 | -9580.52 | -10443.90 |
| 14 | 43 | -6243.69 | -9574.63 | -10438.00 |
| 15 | 46 | -6237.97 | -9571.49 | -10432.00 |
| 16 | 49 | -6234.42 | -9565.60 | -10426.30 |
| 17 | 52 | -6228.51 | -9559.61 | -10420.40 |
| 18 | 55 | -6222.66 | -9554.36 | -10414.40 |
| 19 | 58 | -6217.65 | -9548.41 | -10408.40 |

Three similar cases were simulated with slightly different values for the measurement noise variance. The identified order of the system in all of these cases was order 8 which was in fact the true system order. The values of the AIC for these three cases is shown in Table 3. Note that the AIC falls sharply until order 8 and then slowly increases at a linear rate proportional to the number of additional parameters needed for each increase of 1 in the state order.

### 3.2.1 Beam Model

A finite element analysis was conducted on a homogeneous beam, with both ends unattached, the so-called "Free-Free" Beam Model. This model is potentially very much more complicated than the disk-torsion bar model above, due both to its dynamics and to the ability to create a very large system order. We will briefly describe development of a 5- segment version of this beam, and then a 2-segment version. The latter is almost too coarse to be useful, but it was necessary to start at this level due to the strain on computing resources which the 5-segment version creates.

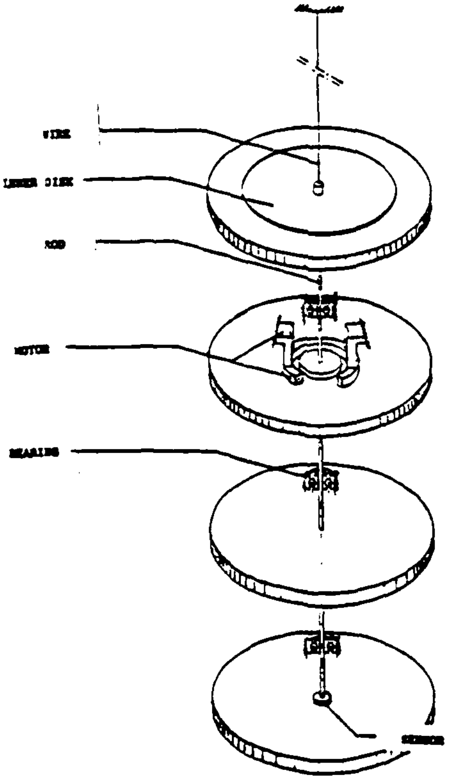


Figure 4. Laboratory Four-Disk System

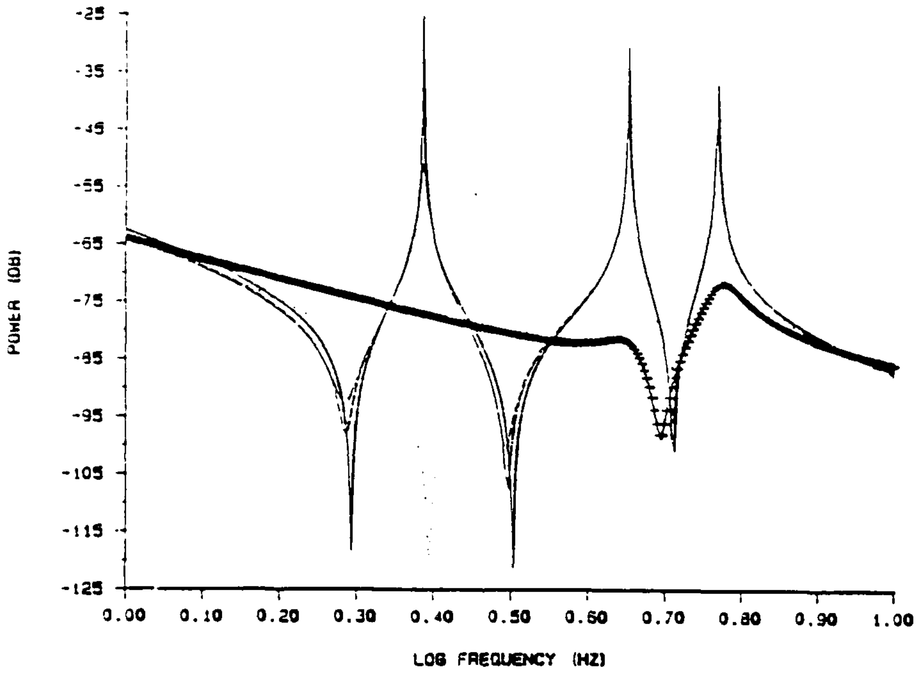


Figure 5. Transfer Function Frequency Response Magnitude
( - true model; identified models: + 0.05, ··· 0.005, ⎯ ⎯ 0.0005, — 0.0 )

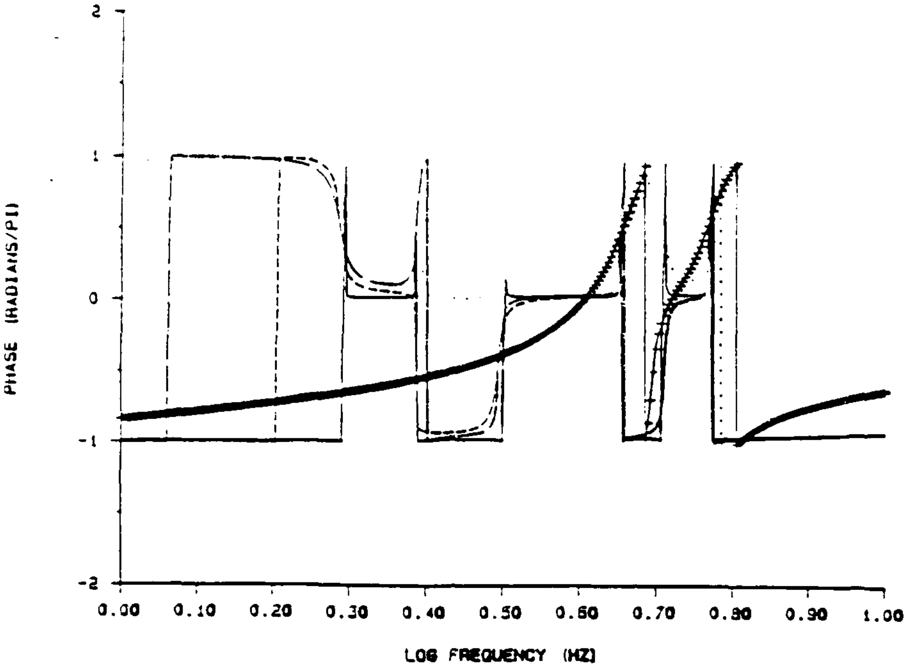


Figure 6. Transfer Function Frequency Response Phase
( - true model; identified models: + 0.05, ··· 0.005, _._ 0.0005, — 0.0 )

The general equation for transverse deflection of a beam is given by

$$-\frac{\partial^2}{\partial x^2}\left[EI(x)\ \frac{\partial^2 y(x,t)}{\partial x^2}\right] + f(x,t) = m(x)\frac{\partial^2 y(x,t)}{\partial t^2} \tag{40}$$

where $f(x,t)$ is the applied force, $t$ is time, $x$ is axial direction, and $y$ is beam deflection from the axis. Applying the usual solution methods for the free-free boundary condition case leads to the following expressions for (symmetric) elemental mass and stiffness matrices:

$$M_j = \frac{m_j h_j}{420}\begin{bmatrix} 156 & 22h & 54 & -13h \\ & 4h^2 & 13h & -3h^2 \\ & & 156 & -22h \\ & & & 4h^2 \end{bmatrix}$$

$$K_j = k_j\begin{bmatrix} 12 & 6h & -12 & 6h \\ & 4h^2 & -6h & 2h^2 \\ & & 12 & -6h \\ & & & 4h^2 \end{bmatrix}$$

where

$$k_j = \frac{EI}{h_j^3}$$

$E$ is Young's modulus, $I$ is beam inertia, and $h_j$ is the segment length.

The finite element model segments each have 2 nodes, and at each node the transverse displacements are expressed as a translation and a rotation (Figure 7a). The basic 5-segment model developed for LSSICS analysis consists of equal length segments and 6 nodes. The displacement vector is this of dimension 12 (Figure 7b).

The dynamic relation (39) then becomes

$$M\ddot{u} + D\dot{u} + Ku = f \tag{41}$$

If we define $q$ as the state variable vector of modal deflections, Equation (40) converts to

$$\dot{q} = \begin{bmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{bmatrix} q + \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix} f \tag{42}$$

where

$$q \triangleq \begin{bmatrix} \dot{u} \\ u \end{bmatrix} \tag{43}$$

and $D, M$ and $K$ are built from the elemental matrices $M_j$ and $K_j$ in the usual way. Except where indicated, we assume no damping ($D = 0$). Note that $q$ has 24 elements, and that the submatrices in Equation (41) are 12-by-12.

This model is currently so large for efficient analysis of the CVA identification algorithm that a smaller, 2-segment model has been developed. The vector $q$ in this case has dimension 12. A 2-input, 2-output system has been defined by sensing translation and rotation deflections only at node 3 ($u_5$ and $u_6$), and placing actuators at node 2 ($u_3$ and $u_4$).

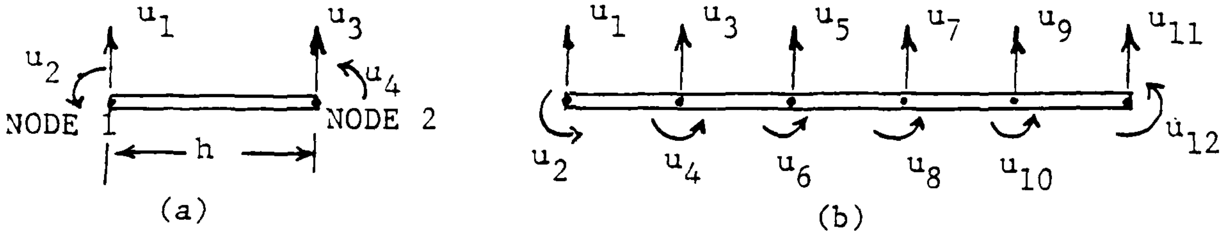


Figure 7. Displacement Variable Definition for 5-Segment Free- Free Beam

### 3.2.2 Results

The first set of results for the application of CVA identification to flexible beam dynamic systems is presented for the case of the symmetric 2-segment beam, derived from the left two segments of the beam pictured in Figure 7b. The dynamic model to be identified is given by Equation (41), and the same 2-input, 2-output formulation as defined above is also used here.

The results are shown in Figure 8 in the form of transfer function magnitude, magnitude squared ("power"), and phase plots *vs.* frequency. In this formulation, there are four total transfer functions between all combinations ot the 2 inputs and 2 outputs. They are numbered like the four elements of a 2-by-2 matrix, with element $(i,j)$ connecting input $j$ to output $i$. The plots in Figure 8 are representative, and do not, for example, show all of the phase plots.

The true transfer function has two poles influenced by the translational force input $u_3$, and another two poles influenced by the torque input $u_4$. The rigid body modes of course occur at zero frequency.

Analysis of the data shows that CVA does a good job of locating the frequencies of vibration, but does less well in isolating totally the response of the force input, (.,1) to the three nodes of the true system. That is, the poles associated with the torque input (.,2), appear both in the torque response, (2,.), and the force response, (1,.). This effect is best seen in the transfer function magnitude plots of Figure 8; the transfer function squared plots tend to obscure this effect, and hence highlight the good agreement between the true and estimated responses. The representative phase plots in Figure 8 are more accurate than appears, due to the "wrap-around" effects at $\pm\pi$.

Table 4 shows that the identified basic system dynamic modes match those of the truth system rather well (the discrete system eigenvalues are shown here; continuous system comparisons would be equivalent, given knowledge of the sampling interval).

The coupling between the torque inputs and force outputs mentioned above is felt to be due to the unrealistically high degree of symmetry of the original two element beam model, with the control inputs located exactly in the center of the beam. We thus analyzed the affects of asymmetry by developing a model whose central node is at the 60% point along the beam. Figure 9 shows that the asymmetry greatly enhances the accuracy of the identification. Plots (a) and (b) of this figure even show fine agreement at the smaller frequencies in a log plot, which is very difficult to achieve, since the statistical "strength" of an identification is spread more or less evenly over the frequency, and not its log.

Plots (c) through (r) in Figure 9 plot the (a) and (b) results on the same frequency scale used in Figure 8. Here, there is no need to use the square of the transfer function to highlight the similarities. All of the transfer functions agree very well with the true system transfer functions, both in magnitude and phase, as the other plots in Figure 9 show. Table 5 shows that the modes for the asymmetric two segment beam are also very accurately identified. The symmetric beam identification does at least as well in this aspect, as discussed above, which is the significant information needed for adaptive control.

Identification was also done on a "51%" beam, with results being clearly better than the symmetric case, but not quite as good as the "60%" case. The conclusion is that some asymmetry is beneficial to some aspects of the overall identification process, but is not critical to achieving good real time control.

Finally, effects of measurement noise on the identification are analyzed. The "60%" beam was used to generate the results shown in Figure 10. Plot (a) represents the case $cov(R) = diag(5.8E-06, 1.2E-05)$, where $R$ is the symmetric measurement noise covariance matrix (off-diagonal terms not given). Plot (b) represents the case $cov(R) = diag(0.0020, 0.0035)$. Each of these plots compares directly with Figure 9a,

Table 4: System Eigenvalues, 2 Segment Beam

| Identified System | True System |
|---|---|
| 0.7164372 ± $i$0.6976606 | 0.7164396 ± $i$0.6976491 |
| 0.8854587 ± $i$0.4646908 | 0.8854687 ± $i$0.4646990 |
| 0.9813648 ± $i$0.1920686 | 0.9813784 ± $i$0.1920845 |
| 0.9980200 ± $i$0.6102011E-01 | 0.9980949 ± $i$0.6169782E-01 |
| 0.9994899 ± $i$0.6144221E-03 | 0.9999998 ± $i$0.6324555E-03 |
| 1.0017200 ± $i$0.1096435E-02 | 1.0017510 |
|  | 0.9982523 |

Table 5: System Eigenvalues, 2 Segment Asymmetric Beam

| Identified System | True System |
|---|---|
| 0.5922647 ± $i$0.8057591 | 0.5921894 ± $i$0.8057988 |
| 0.9010977 ± $i$0.4336092 | 0.9010917 ± $i$0.4336285 |
| 0.9796414 ± $i$0.1995879 | 0.9798446 ± $i$0.1997612 |
| 0.9975095 ± $i$0.6225727E-01 | 0.9980766 ± $i$0.6199332E-01 |
| 0.9930746 ± $i$0.1769323E-01 | 0.9999997 ± $i$0.8094876E-03 |
| 1.0011140 ± $i$0.2105826E-02 | 1.0001180 |
|  | 0.9998822 |

for which there was no measurement noise. It is clear that "modest" levels of noise (Figure 10a) are acceptable, but that at some point the identification becomes too difficult (Figure 10b). It is felt that the noise level in Figure 10a is well within state-of-the-art capabilities for LSS applications.

## 3.3 Model Predictive Control Results

Three flexible structure models were used in analyzing the applicability of MPC to LSS control. These are: the disk- torsion bar model of Cannon and Rosenthal (1984), using the parameter values in Tahk and Speyer (1987); and the 2- and 5- segment finite element models of a "free-free" beam, discussed in the previous section.

It is tempting to dismiss the relatively simpler disk-torsion bar model from the analysis; however, only this model is of a size which allows us to demonstrate the stability analysis features of MPC. We emphasize again that our Phase II activities will be supported by appropriate simulation resources for a complete analysis.

The disk-torsion bar model, for the given parameter values which include light damping, has a magnitude and phase spectrum as shown in Figure 11. This corresponds to a SISO system, with collocated sensor and actuator at mass 2; the other masses are totally free to move.

The key MPC design parameters are: $T_d$, system quantization interval (seconds), $\tau$, the vector of reference trajectory time constants (seconds), error and control weights $Q$ and $R$, smoothing points per control cycle, NSM, and number of control updates in the horizon of prediction, $L$.

Figure 12 shows the result of the MPC-controlled disk-torsion bar system when mass 2 is commanded to slew 2 "units" from a quiescent state. In this as in all time response plots, the abscissa shows time in seconds. For relatively little design effort, a very effective result has been obtained. The mass is controlled to its desired position to within 1% error, after reaching this level in under a second. Note that the uncontrolled masses still exhibit very lightly damped, oscillatory responses, which require control response due to coupling. This "tight" solution results from zero control weighting ($R = 0$). A tradeoff is set up between tight response and control energy required for same, as seen in Figure 13, which shows results for $R = 0.1$ and 1.0. Note the expected result of large demands on an unweighted control input. This study highlights the ability of MPC to control specific structural nodes.

Following the closed loop stability analysis approach outlined in Section 2.3, the appropriate $F_{cl}$ and $G_{cl}$ matrices were created. Figure 14 shows how the closed loop transfer function for $R = 0$ is superior to both the $R = 1.0$ and open loop case (latter in Figure 11).

An offline design procedure may clearly be developed based on $F_{cl}$ and $G_{cl}$ using standard analysis tools. Since these matrices are explicit functions of the MPC design parameters, the latter may be varied until curves such as Figure 14a result. This analysis, however, is very burdensome on the core available to a PC or even a $\mu$-VAX; we propose in Phase II to use the stability matrices as both design and analysis tools on larger systems, when Grumman's facilities are available.

This investigation will also focus on ways to generate $F_{cl}$ and $G_{cl}$ more efficiently. For example, if $Q$ is purely diagonal - the only case so far analyzed - there is great potential for very large savings in core, since $(p \cdot \text{NSM} \cdot L)(p \cdot \text{NSM} \cdot L - 1)$ locations become superfluous.

A final note: it is apparent that the closed loop stability matrices $F_{cl}$ and $G_{cl}$ can be used to simulate the MPC closed loop response. Doing so produces a response which, as expected, matches perfectly the Figure 12 response.

### 3.3.1 Free-Free Beam Design

Analysis began first with the 2- segment finite element model of the free-free beam described in Section 3.2. The same 2-by-2 input/output structure defined in that section is used here as well. This is an exceptionally difficult system to control, for these reasons:

(i) sensors and controls are not collocated

(ii) usually, every displacement state is controlled directly; in this case, only 2 controls are applied to 6 displacement states

(iii) the basic model has no damping (damping is supplied in some cases, however)

Item (iii) is emphasized because it makes the open loop system technically unstable. The "difficulties" imposed are partly due to a desire to severely test MPC, but also to core limitations.

Figure 15 shows the open loop (uncontrolled) response of the undamped 2-segment model to a deflected initial state, no controls. This same initial state is used for all examples in this section. This is a regulator, as opposed to tracking problem. The four "non-output" responses represent the unmeasured nodal deflection variables, consisting of two translation and two rotation variables. None of the 6 deflection *rate* variables are considered for measurement or control in this analysis, although this is clearly a key item for future work.

One of the best MPC designs to date, which is a clear improvement over the uncontrolled case, is shown in Figure 16. The fact that control activity has lessened considerably after 5 seconds while not completely eliminating response oscillations - though they appear to be slowly damping out - indicates that this particular sensor/actuator structure is not optimally suited for control. Notwithstanding this observation, and in view of the three design difficulties imposed on MPC by this particular problem, MPC clearly improves performance. For this case, $R = 0$, NSM=12, and $L = 8$, and the weight $Q$ is a pure diagonal whose elements monotonically increase from upper left to lower right on the diagonal. When the control prediction window $L$ is increased to 12 in this problem, the results are better still, as shown in Figure 17.

More than the other MPC design parameters, it seems that NSM, $L$, and $R$ have the most direct effect on a good MPC design. Halving $T_d$ to 0.025 actually had a destabilizing effect on performance, as seen in Figure 18. The parameter $r$ was increased by a factor of 10 to 1.50 seconds in another study, but this had very little effect on response. Also, as pointed out by Reid et al., (1981), setting $R = 0$ usually results in tighter response. This is of course preferred as long as there is sufficient control energy. We noted the same effect in the disk-torsion bar problem discussed above.

System damping remains an effective means of reducing the system's internal energy and with it, the stabilizing control effort required. Figure 19 shows a sequence of MPC solutions for a system in which damping increases from none (a) to less than about 0.1% (b), to about 4% (c). Solution (a) represents the first 3 seconds from the more complete solution in Figure 16. (Economy of simulation time is very important: each solution in Figure 19 takes over an hour to generate on a PC; the longer one in Figure 16 takes about 6 hours.) Note that with even the addition of the most mild damping, the control effort decreases much more than the output response differs. It is not until rather significant damping is added (Figure 19c) do the outputs change markedly. The open loop 4% damping case is shown in Figure 21 for comparison with Figure 19c. See also the plots for the unmeasured ("non-output") variables in Figure 20. Damping greatly affects their amplitude. This analysis supports Reid's rule of thumb that NSM and $L$ each be at least of order $n$.

The next series of simulations deal also with the two-segment beam model, but without damping. The above analysis was deliberately set up to be very difficult on a control algorithm; there were much fewer inputs and outputs configured than there are system modes, and the basic translational and rotational displacement and velocity coordinates which have been used generate a high degree of numerically inefficient coupling. It was felt that transforming the physical coordinates to the "eigen", or modal, coordinates would provide MPC a better environment for a good design.

The transformation is performed on the dynamic system (41) (with $D = 0$ here), and converted into the expression

$$\ddot{\eta} + \Lambda\eta = Y^T f \tag{44}$$

where $Y = Q^{-1}U$, $Q$ is the Cholesky factor of the mass matrix, $M = Q^T Q$, and $U$ is the unitary matrix resulting from applying the Schur decomposition to $\overline{K} = Q^{-T} K Q^{-1}$, i.e., $U \Lambda U^T = \overline{K}$. Initially, the output equation was retained, as follows:

$$y = Cq = CY\tilde{\eta} \tag{45}$$

where $\tilde{\eta}$ is comprised of $\eta$ and $\dot{\eta}$ in the same manner as $q$ in Equation (43).

The Figure 17 case was rerun using the modal formulation just described. The results, shown in Figure 22, clearly show significant improvement resulting from the modal formulation. We emphasize again that this is a system with 6 dynamic modes, subject only to 2 remote actuators and 2 colocated sensors.

Somewhat surprisingly, when the actuators are also colocated at one end of the beam, a poorer design results (Figure 23). This is likely due to a decrease in controllability resulting from the distance of these actuators from the other nodal points. Finally, when four actuators are devoted to the control of the end-beam displacement variable, making a 4 input, 1 output configuration, Figure 24, the result is even better than the case shown in Figure 22 (the four inputs here are at the 3, 4, 5, and 6 positions in Figure 7, and the single output is beam displacement at station 5). This result matches expectations.

An MPC design has been initiated for the 24 order five-segment free-free beam model described in Section 3.2. At the time of this report, this analysis is not yet complete. System order has initially forced smaller values for NSM (4) and $L(7)$ than desirable. To offset this somewhat, the 4 sensors and actuators in this configuration were collocated as shown in Figure 25. There is no damping here. The objective again is to regulate each output to zero from a nonzero initial energy state. The two responses shown in Figure 26 are for the case $Q_i \sim 10(i^2)$ and $R = I$. This value of $R$ is driven less by proper application of design principles learnt thus far than by accommodating the algorithmic numerical deficiencies of the $\mu$-VAX resident Fortran code. At this time, the more numerically stable and core efficient MPC code is on the PC, written in PC-MATLAB. The Fortran code is currently being upgraded.

This examples does point out however, the potential for MPC to control moderately large, lightly damped systems.

### 3.3.2 Robustness Analysis

Robustness is the ability of a control system to perform as optimally as possible when unmodeled changes occur in the system, or when the dynamic environment changes in an unpredictable way. MPC is a control method known in past applications to be rather robust. A quantified analysis of the robustness properties of MPC will be conducted during Phase II.

A short series of simulations designed to investigate the robustness of MPC to specific flexible structures. In one case, using the disk-torsion bar model of Cannon and Rosenthal (1984), a "failure" was generated at 4 seconds into a simulation which in every other respect matched the "nominal" case discussed above (see Figure 12). For this case, the viscous damping between mass 1 and mass 2 was halved. Recall that the sensor and actuator are collocated at mass 2, so that its motion is to be controlled directly. Figure 27 shows that MPC successfully offsets the sudden removal of damping within 2 cycles. Note also that both nominal and "failed" period mass 2 trajectories are controlled to the commanded level to a very high tolerance when the control weight $R$ is set to zero (Figure 27d). If $R$ is nonzero, Figure 28, that is, not well-matched to the problem, the sudden system change can result in an unacceptably oscillatory response.

Robustness of the free-free beam model was analyzed by increasing by 50% the stiffness of the left segment (in the 2-segment case) at 5.0 seconds. Figure 29 shows that MPC again is able to reestablish control.

In both cases, the system or "truth" model was changed but not the control design model. MPC is thus ignorant of the new situation, but is able to perform well. It is useful to note that onboard identification can reestablish for MPC the proper control design model. This is a very important feature of LSSICS; robustness of the "new" configuration is clearly enhanced by accurate knowledge of the system dynamics.

As this phase of the project was ending, some work has begun on the effect of actuator loss (output force/torque = 0 following damage) on system performance. The MPC results discussed above were deliberately generated using fewer control inputs than system modes. A practical design would be more conservative in this area, but this analysis does demonstrate the ability of MPC to stabilize and provide robust performance for "under-controlled" systems. Some comparison was done with Linear Quadratic designs, using full state feedback. Although the results of the comparison are preliminary, the LQ designs seem to be more sensitive in performance to the loss of an actuator. For a baseline LQ design utilizing the full six actuators, failing only one of these at a time, in the two segment free beam case, actually causes instability in four of the six cases. The MPC results discussed above are clearly more robust, although it is recognized that stable LQ designs are possible using only two controls. Further effort is needed in this analysis before firm conclusions can be made.

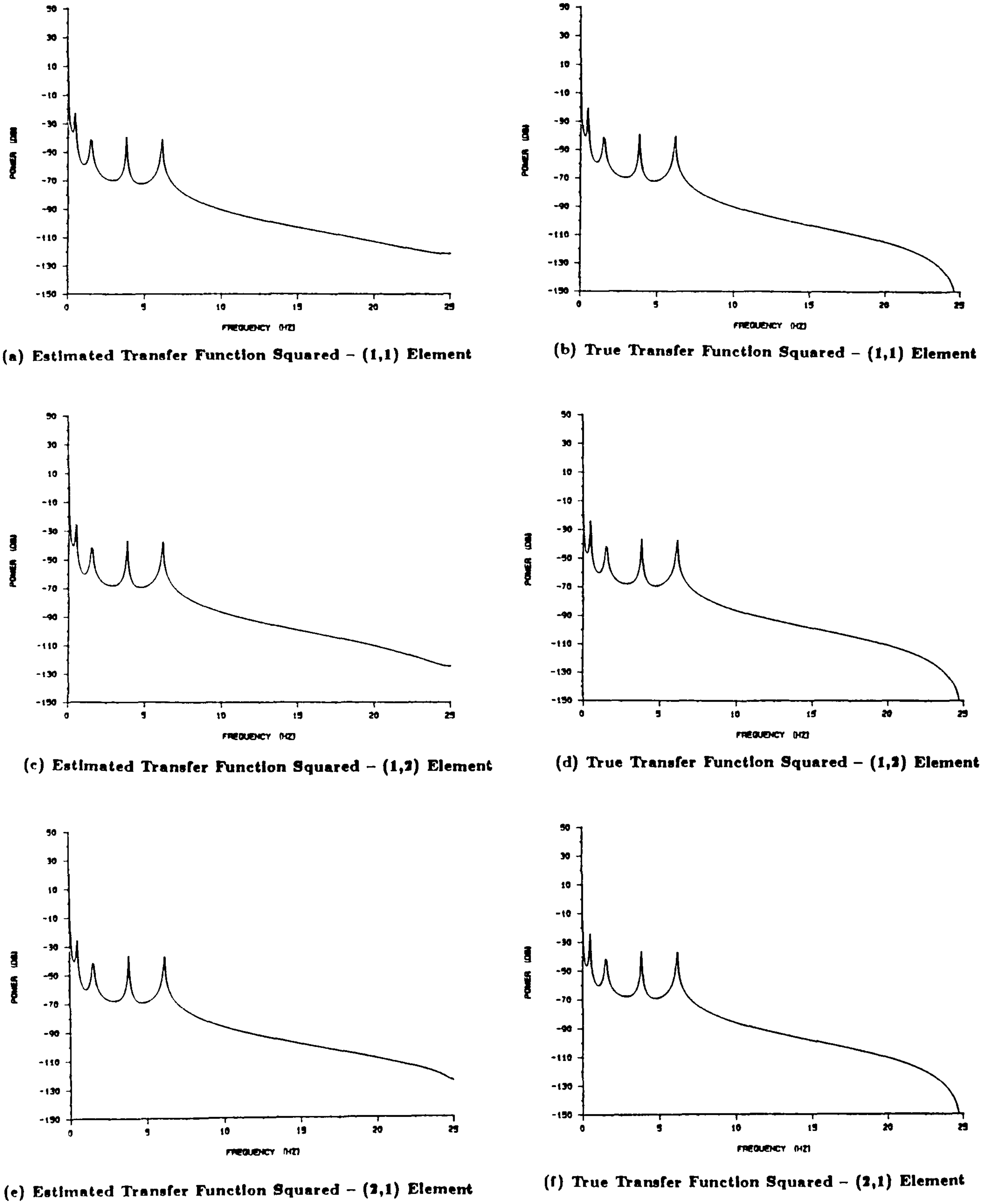
## 3.4 More New Results

This section presents results generated too late for proper location in this report.

A control design scheme called Independent Modal MPC has been developed to enhance the effects and benefits of modal control which were described above. IMMPC operates on the notion that if a system is controllable, it can be controlled by controlling each mode separately. An MPC design is performed on each mode sequentially, and then the resulting controls are combined into a final command input. Figure 30 shows the result when this process was applied once, to the highest frequency mode of the two segment beam. The figure shows the difference in the response of this mode when it is uncontrolled (solid line) and controlled (hatched line). Note that better closed loop damping seems achievable using this approach.

A modal design using IMMPC was next performed configuring sensors of displacement (or rotation) and velocity at each coordinate of the 2 segment beam model. Measurements and control inputs are synchronous at 100 Hz, and the MPC weighting matrices, $Q$ and $R$, were adjusted appropriately for this case. The results (Figures 31 and 32) show that MPC seems to work very well when there is sufficient monitoring of all dynamic modes. Figure 31 presents the uncontrolled (hatched line here, opposite from Figure 30!) and controlled (solid) modal space responses, and Figure 32 the corresponding physical space responses. The length of time in these simulations is very brief, 0.2*sec.*, but appears adequate to show the desired decay of the responses to zero.

It is of course necessary to carry this analysis further by designing reduced order observers, and also by investigating effects of sensor failures. However, it does seem that the basic questions about the applicability of MPC to LSS control are favorably answered.



(a) Estimated Transfer Function Squared – (1,1) Element

(b) True Transfer Function Squared – (1,1) Element

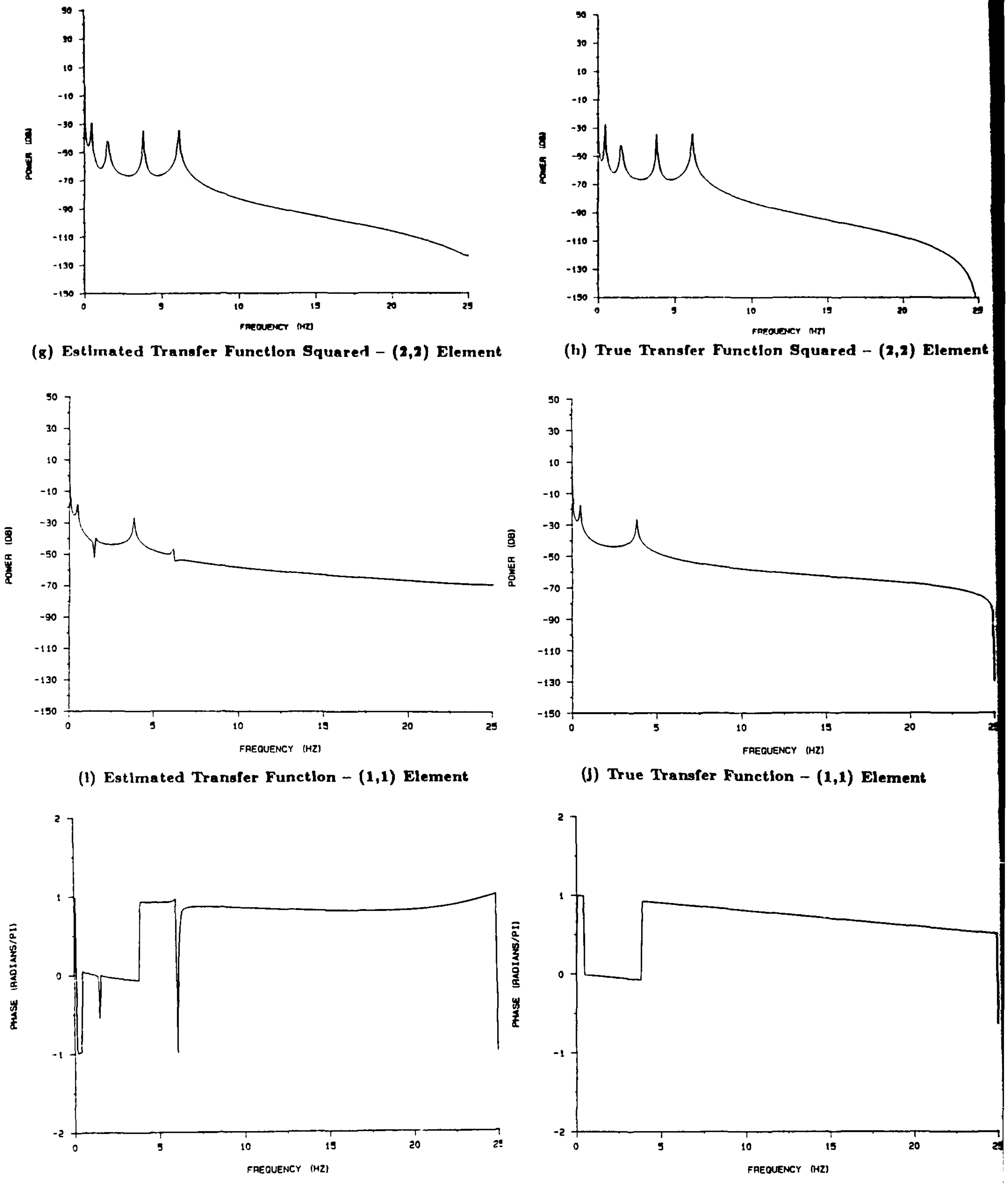
(c) Estimated Transfer Function Squared – (1,2) Element

(d) True Transfer Function Squared – (1,2) Element

(e) Estimated Transfer Function Squared – (2,1) Element

(f) True Transfer Function Squared – (2,1) Element

Figure 8. CVA Analysis of 12th Order Symmetric Free Beam



(g) Estimated Transfer Function Squared – (2,2) Element

(h) True Transfer Function Squared – (2,2) Element

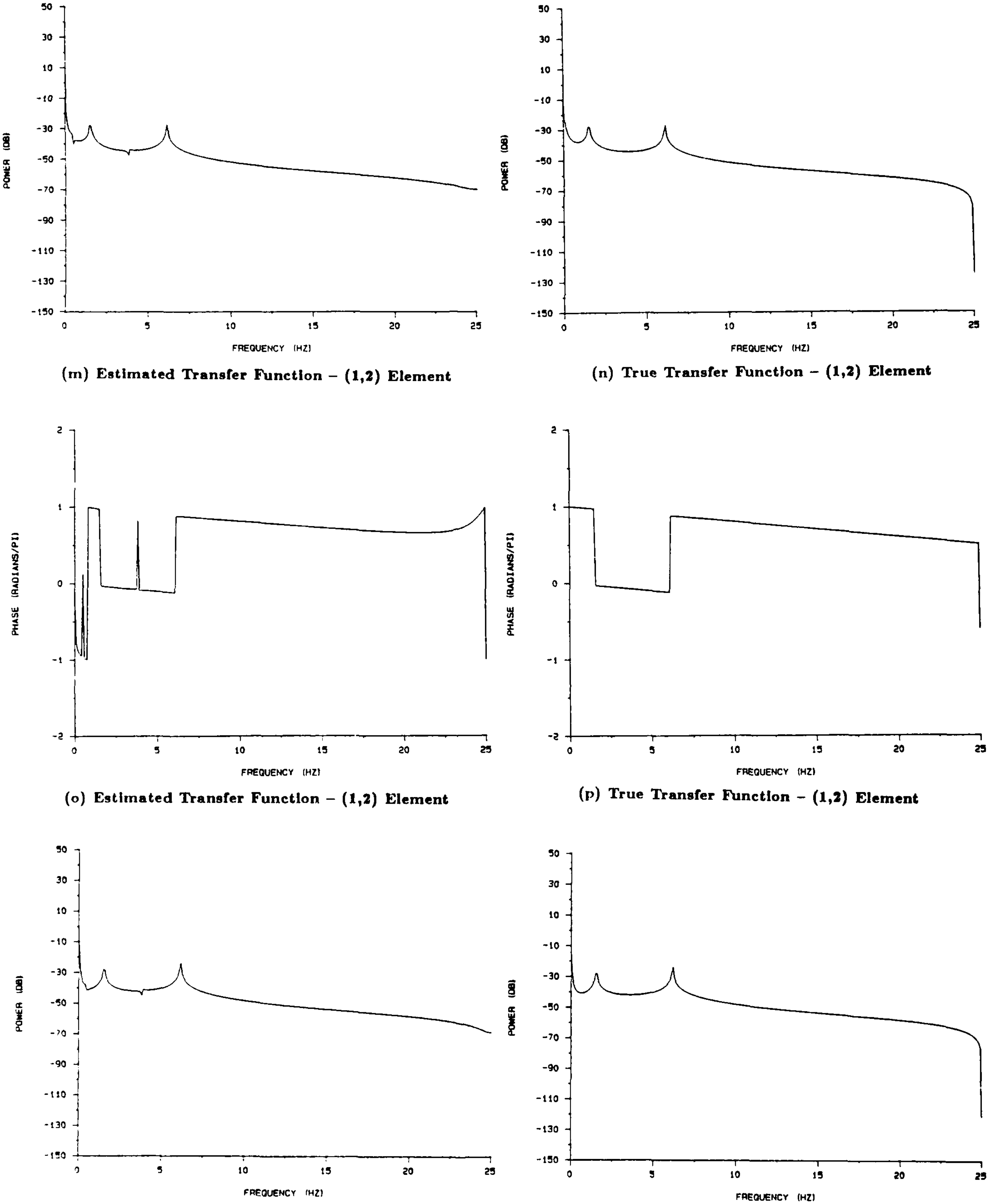
(i) Estimated Transfer Function – (1,1) Element

(j) True Transfer Function – (1,1) Element

(k) Estimated Transfer Function – (1,1) Element

(l) True Transfer Function – (1,1) Element

**Figure 8. CVA Analysis of 12th Order Symmetric Free Beam**



(m) Estimated Transfer Function – (1,2) Element

(n) True Transfer Function – (1,2) Element

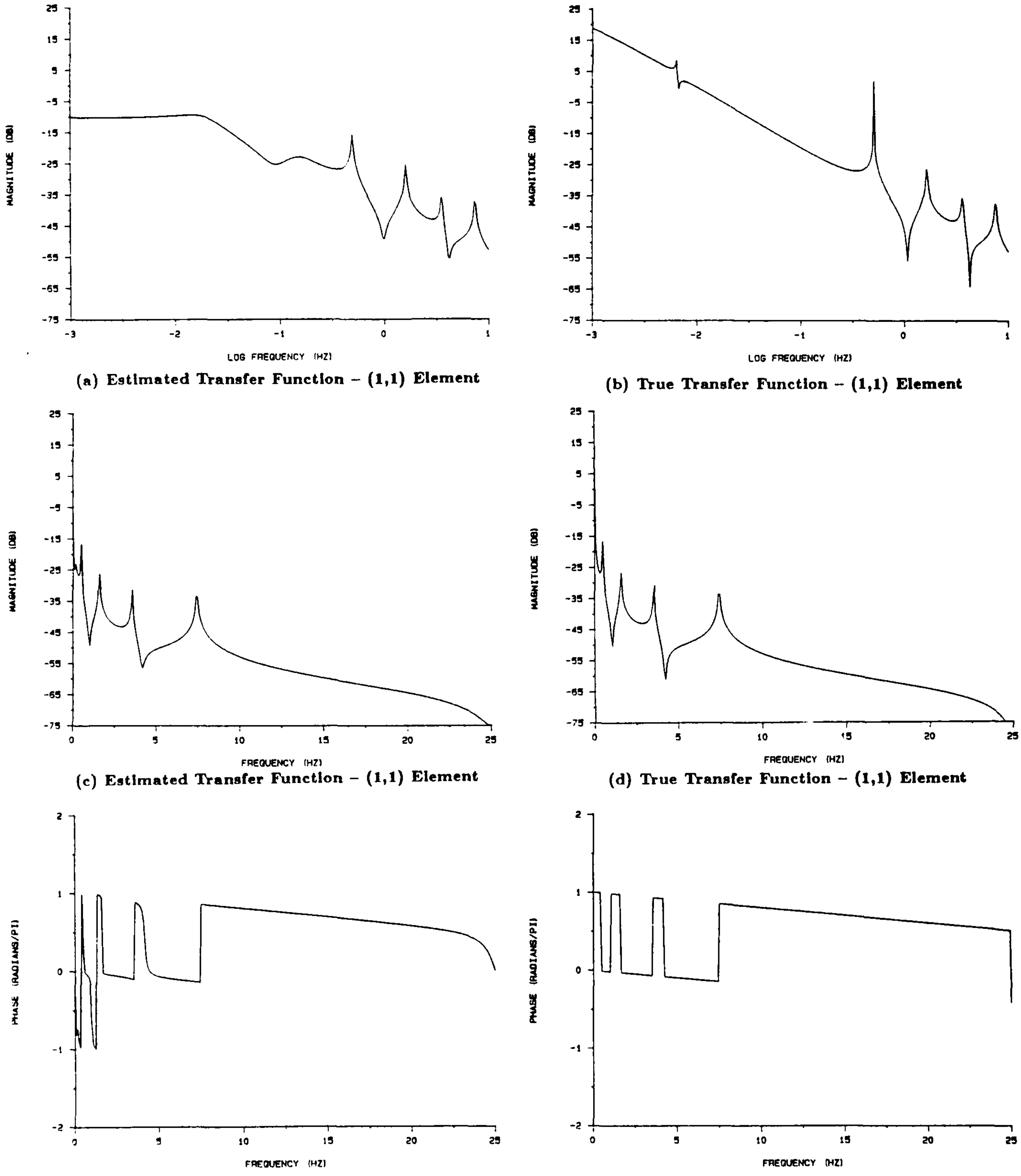
(o) Estimated Transfer Function – (1,2) Element

(p) True Transfer Function – (1,2) Element

(q) Estimated Transfer Function – (2,2) Element

(r) True Transfer Function – (2,2) Element

**Figure 8. CVA Analysis of 12th Order Symmetric Free Beam**



(a) Estimated Transfer Function – (1,1) Element

(b) True Transfer Function – (1,1) Element

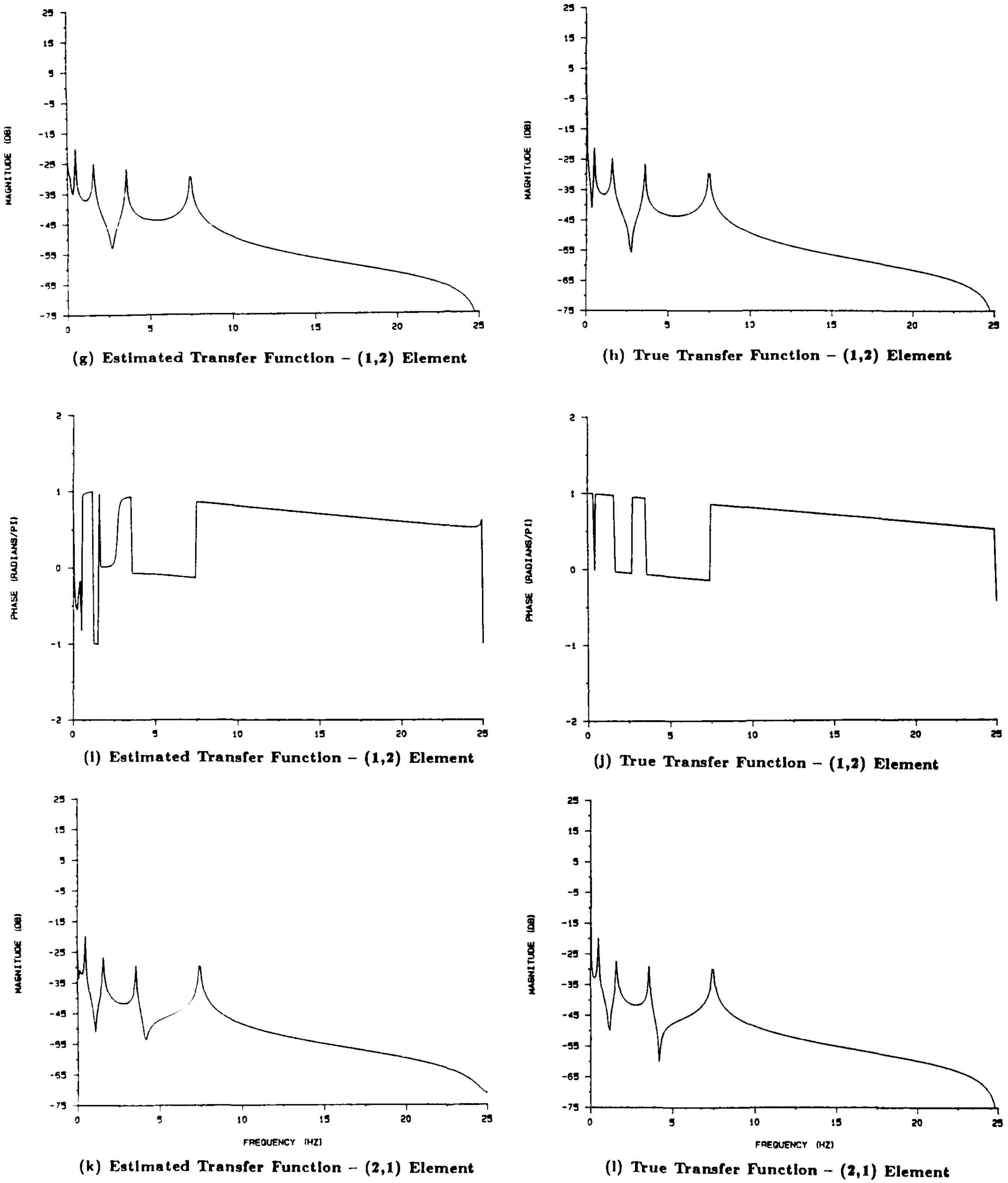
(c) Estimated Transfer Function – (1,1) Element

(d) True Transfer Function – (1,1) Element

(e) Estimated Transfer Function – (1,1) Element

(f) True Transfer Function – (1,1) Element

Figure 9. CVA Analysis of 12th Order Asymmetric Free Beam



(g) Estimated Transfer Function – (1,2) Element

(h) True Transfer Function – (1,2) Element

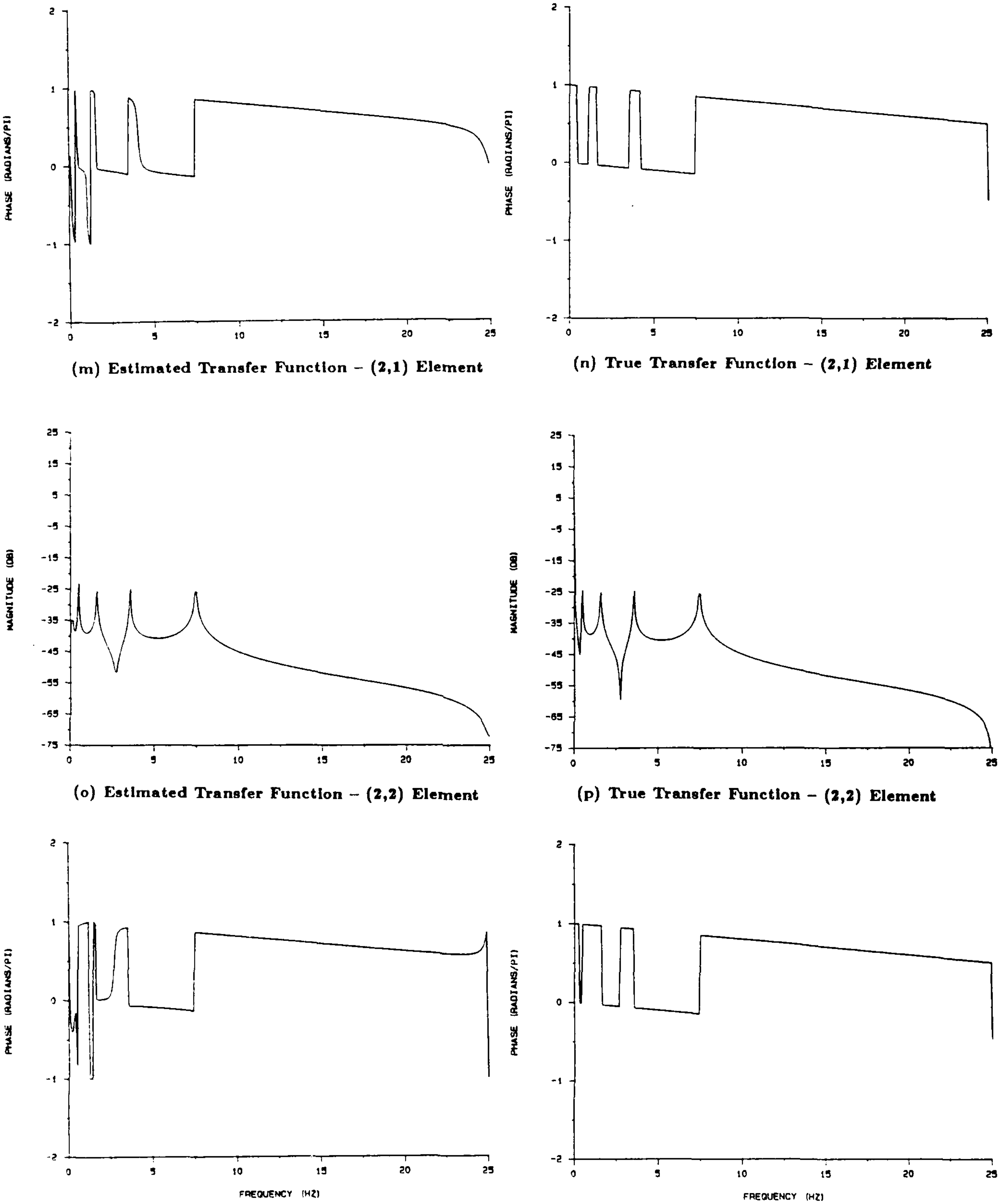
(i) Estimated Transfer Function – (1,2) Element

(j) True Transfer Function – (1,2) Element

(k) Estimated Transfer Function – (2,1) Element

(l) True Transfer Function – (2,1) Element

Figure 9. CVA Analysis of 12th Order Asymmetric Free Beam



(m) Estimated Transfer Function – (2,1) Element

(n) True Transfer Function – (2,1) Element

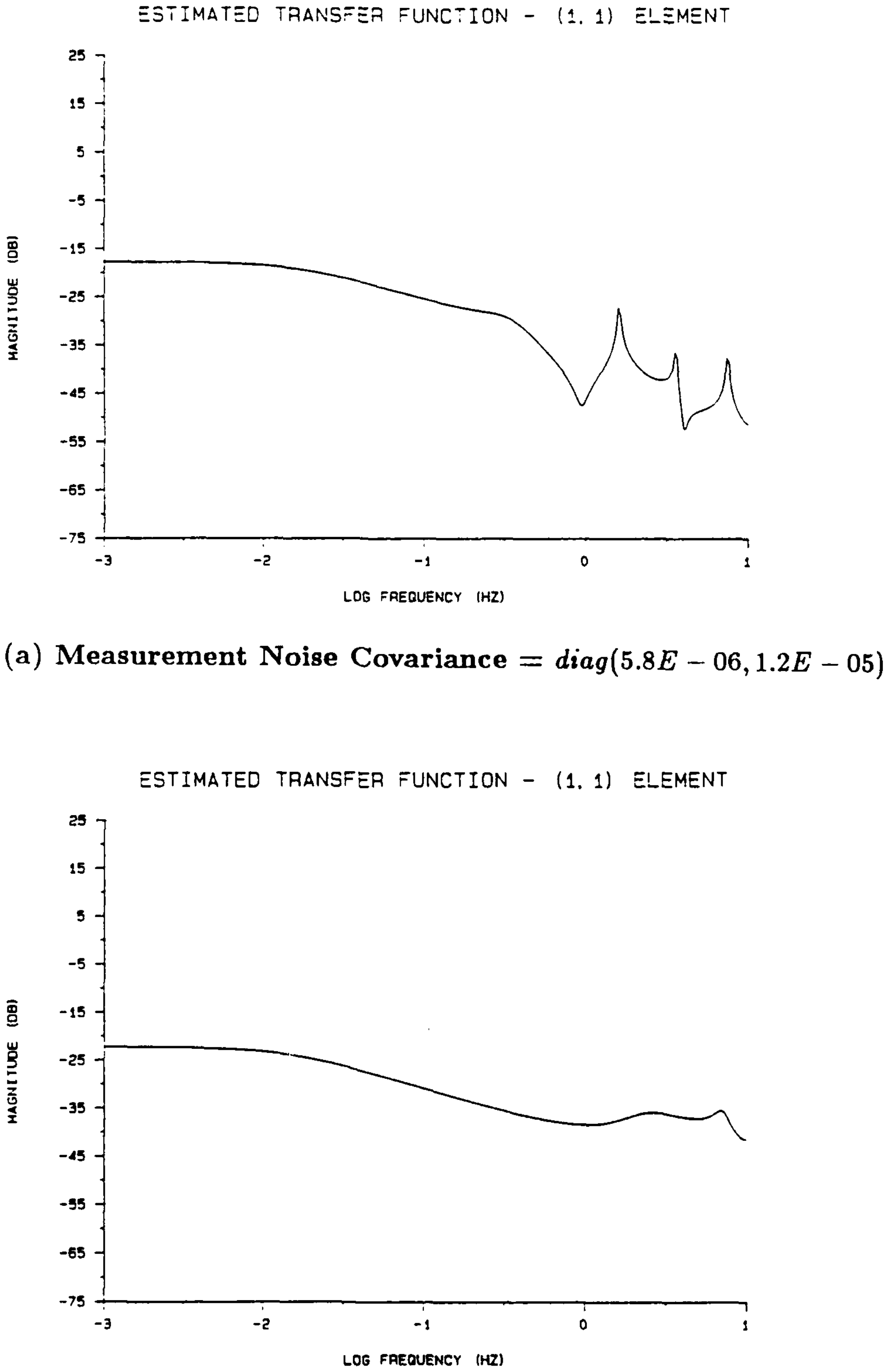
(o) Estimated Transfer Function – (2,2) Element

(p) True Transfer Function – (2,2) Element

(q) Estimated Transfer Function – (2,2) Element

(r) True Transfer Function – (2,2) Element

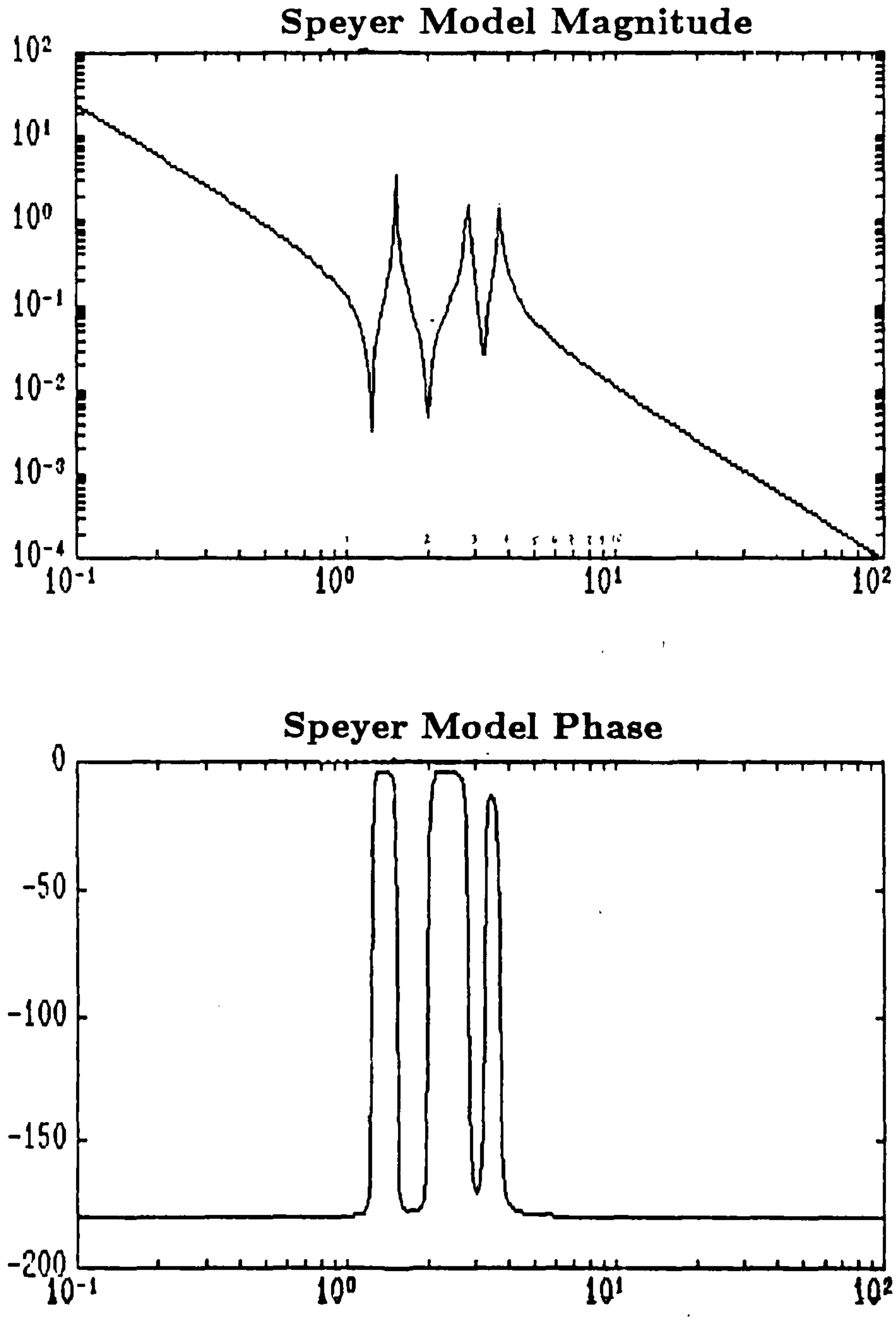
Figure 9. CVA Analysis of 12th Order Asymmetric Free Beam



(a) **Measurement Noise Covariance** = $diag(5.8E-06, 1.2E-05)$

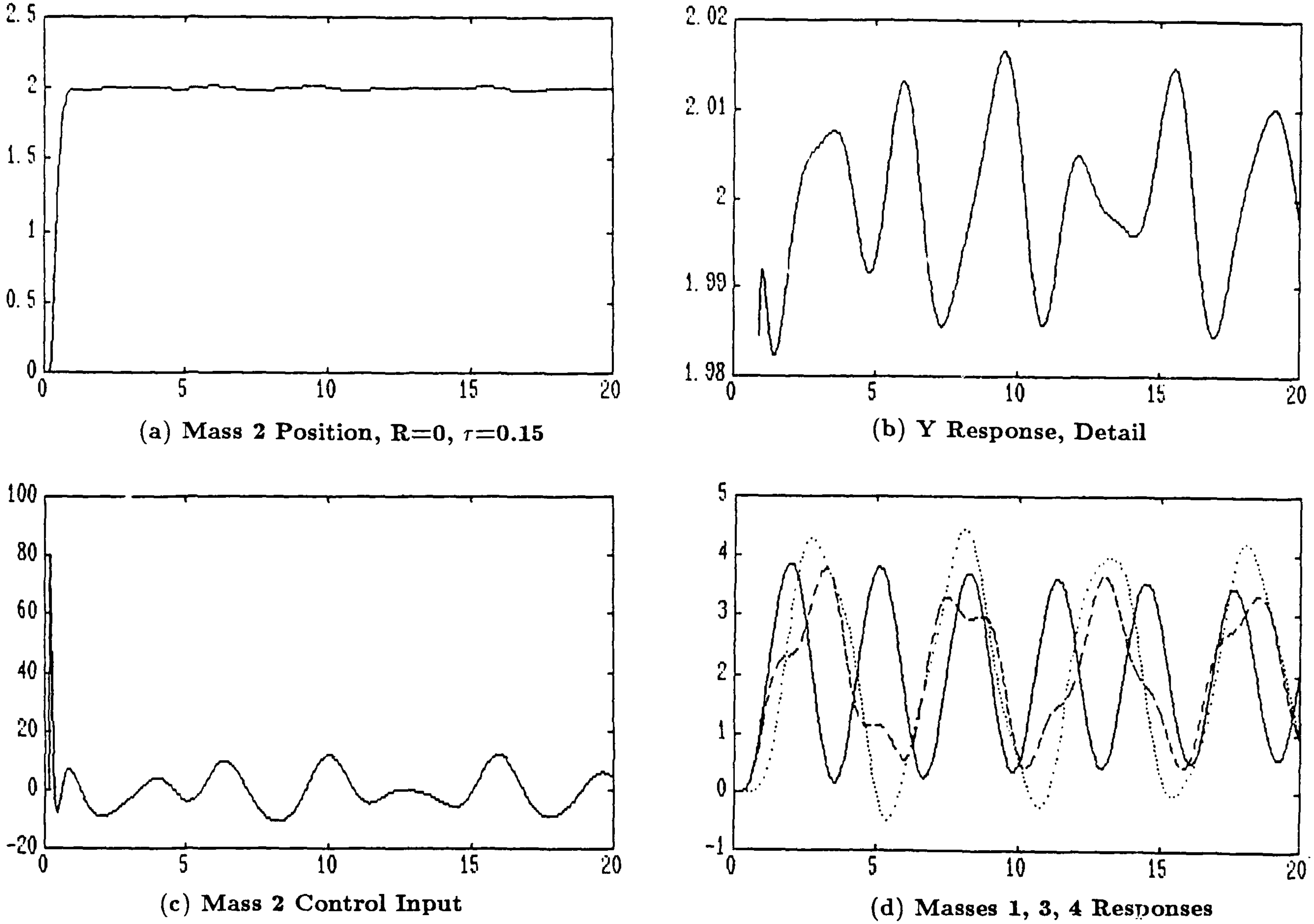
(b) **Measurement Noise Covariance** = $diag(0.0020, 0.0035)$

Figure 10. Effect of Measurement Noise on CVA Identification


Speyer Model Magnitude

Speyer Model Phase

Figure 11. Bode Plots of Disk-Torsion Bar Model



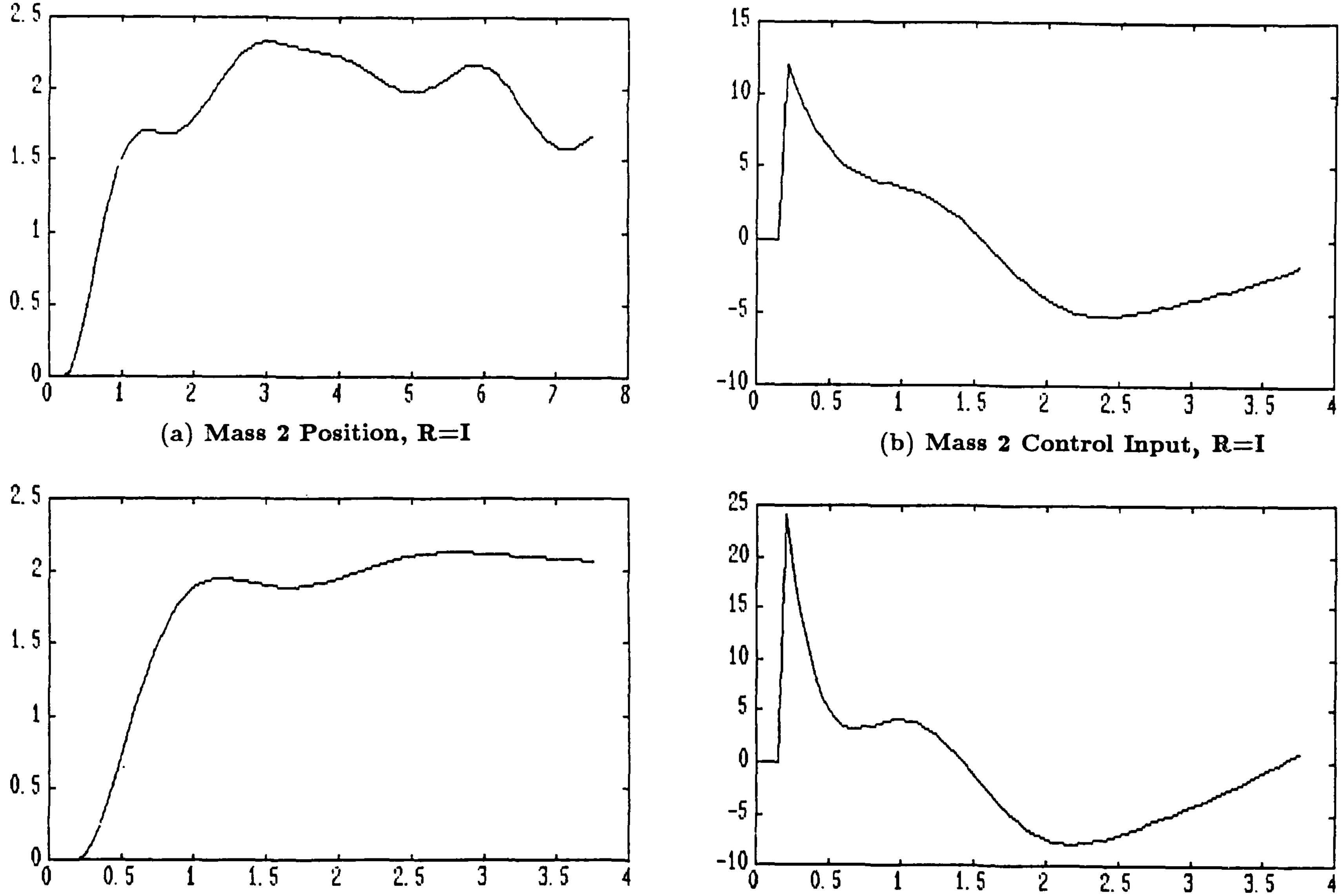
(a) Mass 2 Position, R=0, $\tau$=0.15

(b) Y Response, Detail

(c) Mass 2 Control Input

(d) Masses 1, 3, 4 Responses

Figure 12. Closed Loop Disk-Torsion Bar Model Response to Step Input, Using MPC (Abscissa shows time in seconds).



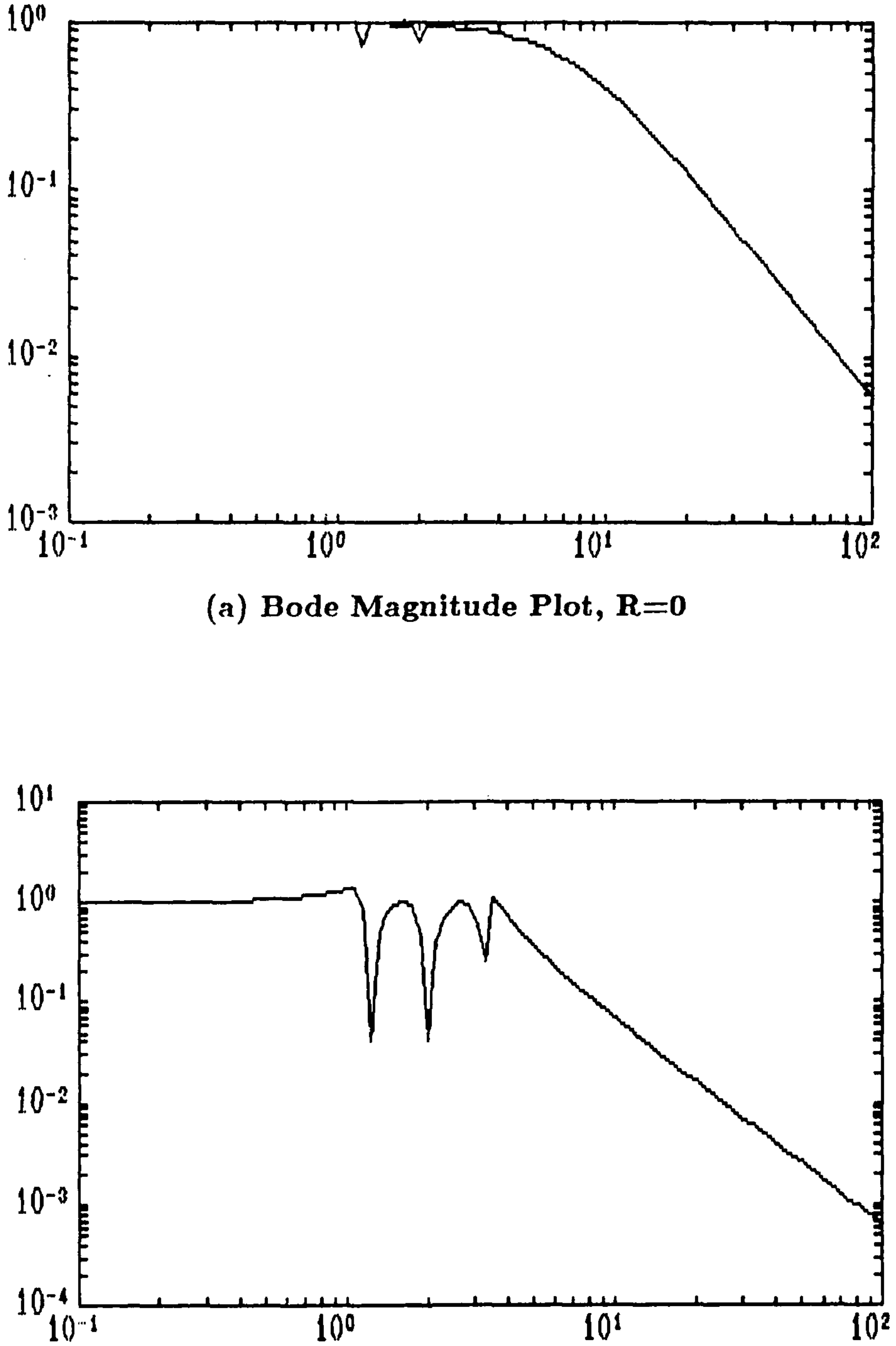
(a) Mass 2 Position, R=I

(b) Mass 2 Control Input, R=I

(c) Mass 2 Position, R=0.1*I

(d) Mass 2 Control Input, R=0.1*I

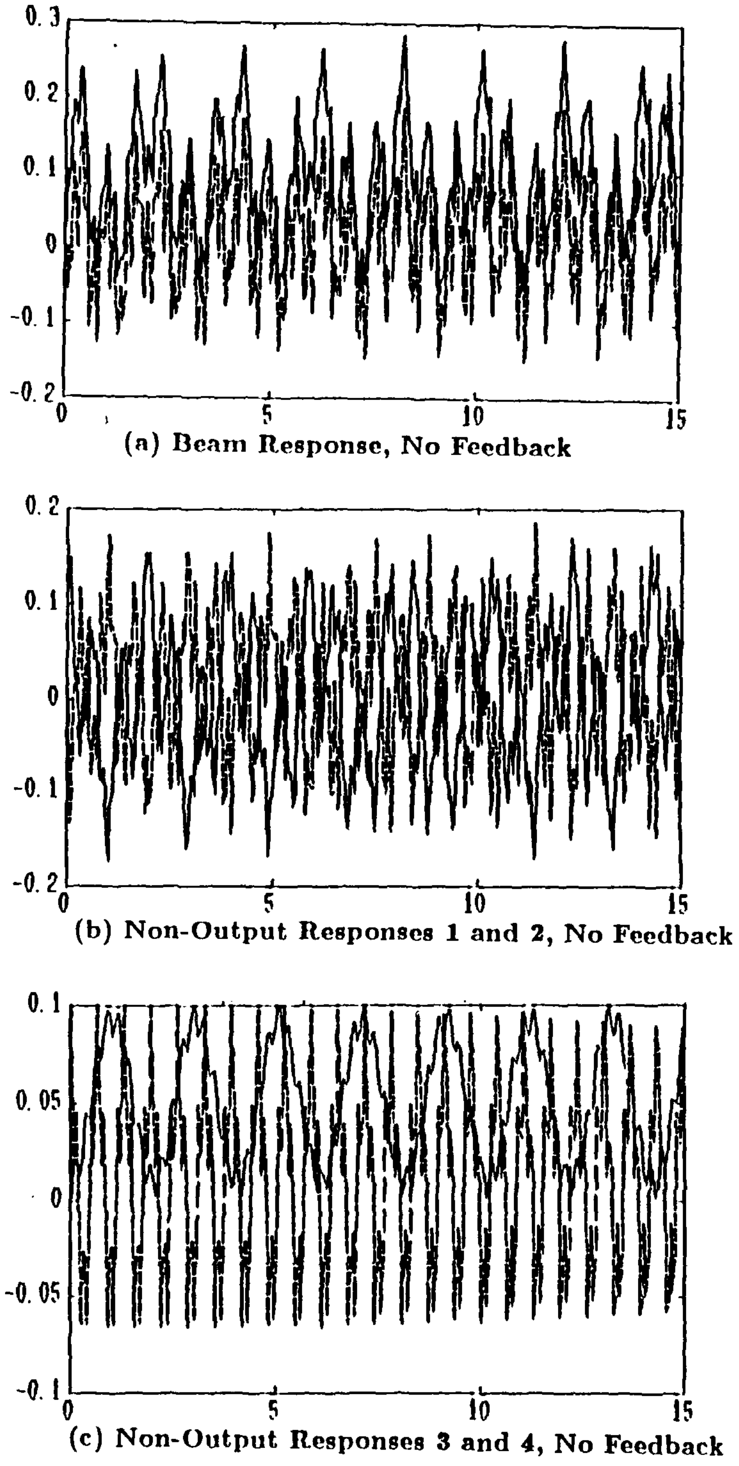
Figure 13. Disk Step Input Response Using MPC



(a) **Bode Magnitude Plot, R=0**

(b) **Bode Magnitude Plot, R=1.0**

Figure 14. MPC Stability Analysis of Disk-Torsion Bar System (a) $R = 0$; (b) $R = 1.0$.

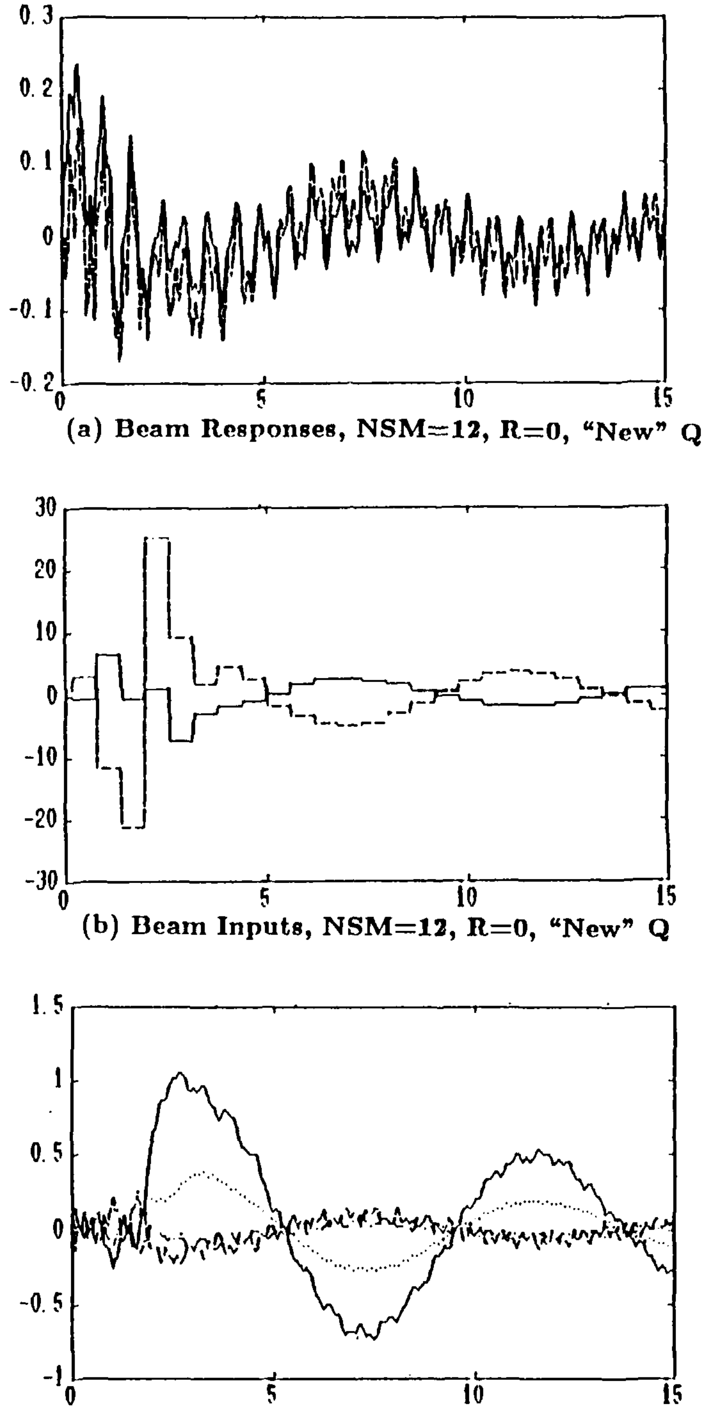


(a) Beam Response, No Feedback

(b) Non-Output Responses 1 and 2, No Feedback

(c) Non-Output Responses 3 and 4, No Feedback

Figure 15. Two Segment Free Beam Open Loop Response

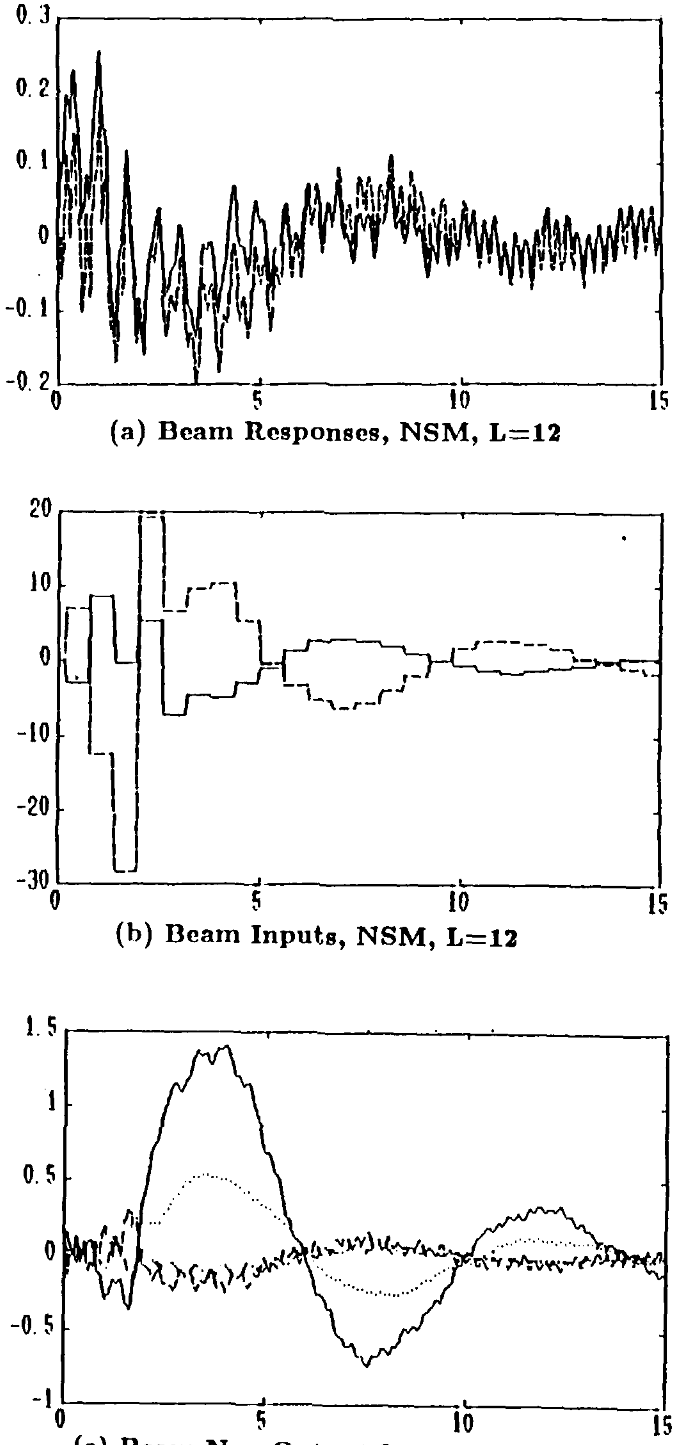


(a) Beam Responses, NSM=12, R=0, "New" Q

(b) Beam Inputs, NSM=12, R=0, "New" Q

(c) Non-Output Responses, NSM=12, R=0, "New" Q

Figure 16. Two Segment Free Beam Closed Loop Response Using MPC; $L = 8$



(a) Beam Responses, NSM, L=12

(b) Beam Inputs, NSM, L=12

(c) Beam Non-Output Responses, NSM, L=12

Figure 17. Two Segment Free Beam Closed Loop Response Using MPC; $L = 12$

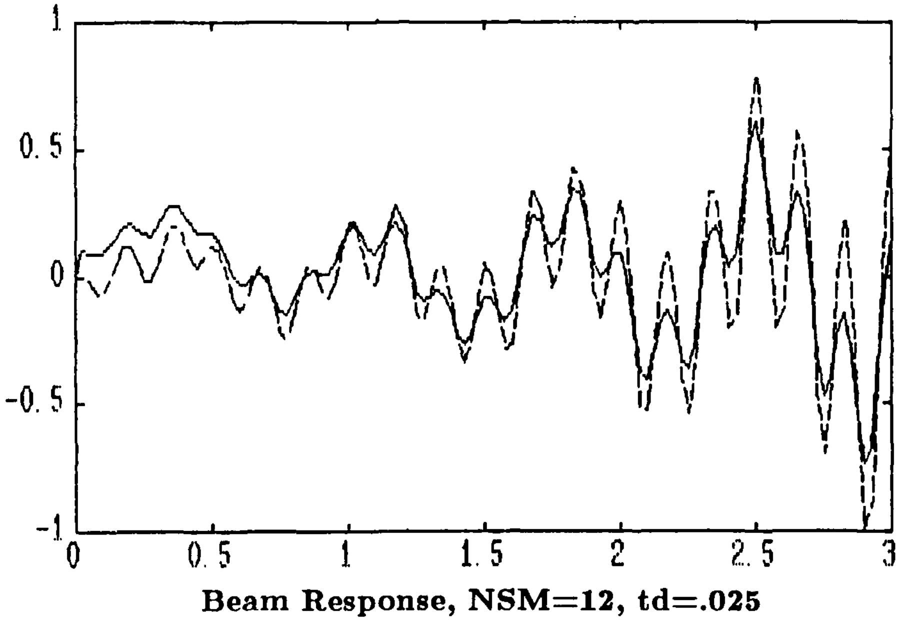
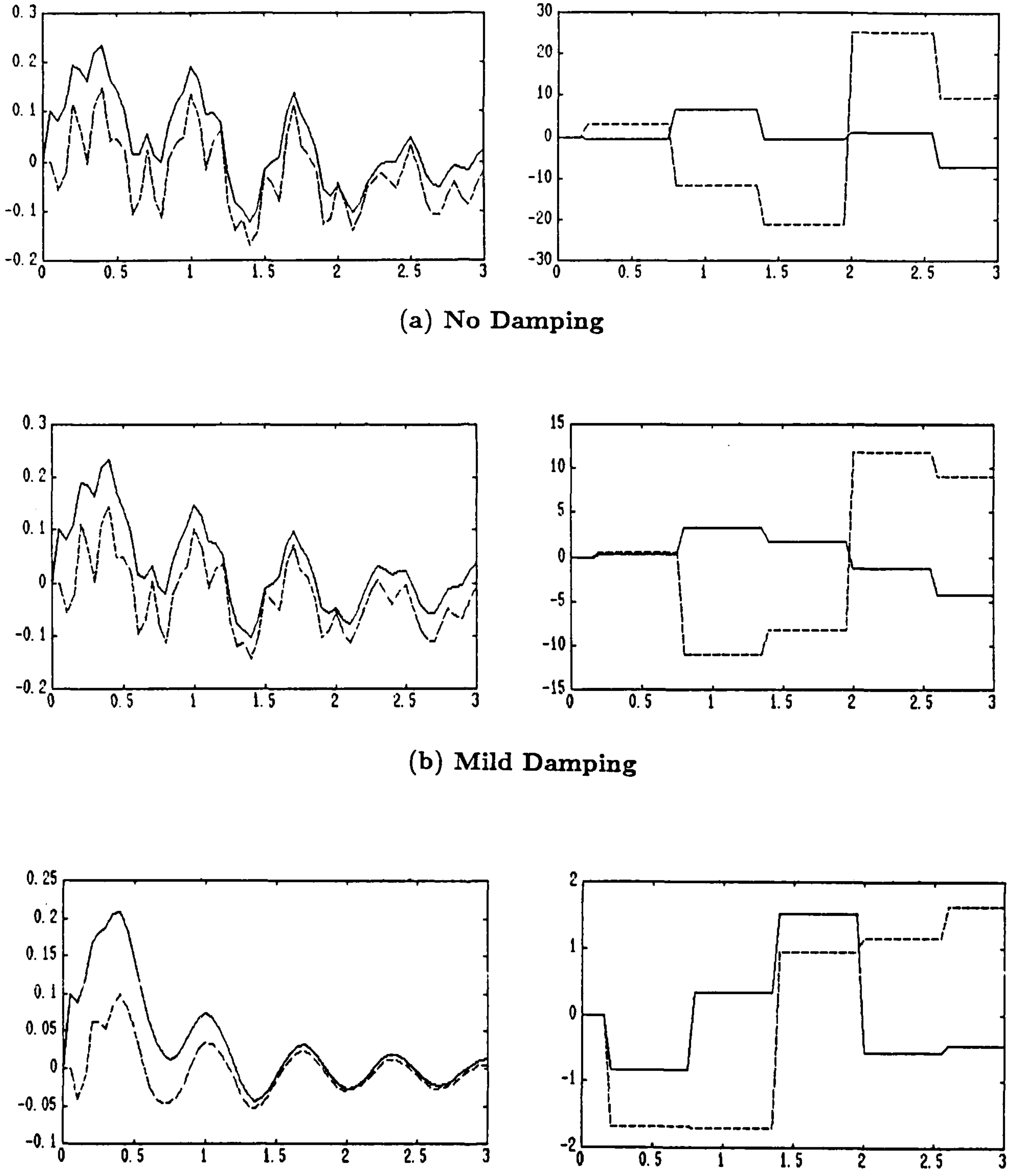


Figure 18. Two Segment Free Beam Closed Loop Response Using MPC; $T_d = 0.025$ sec.

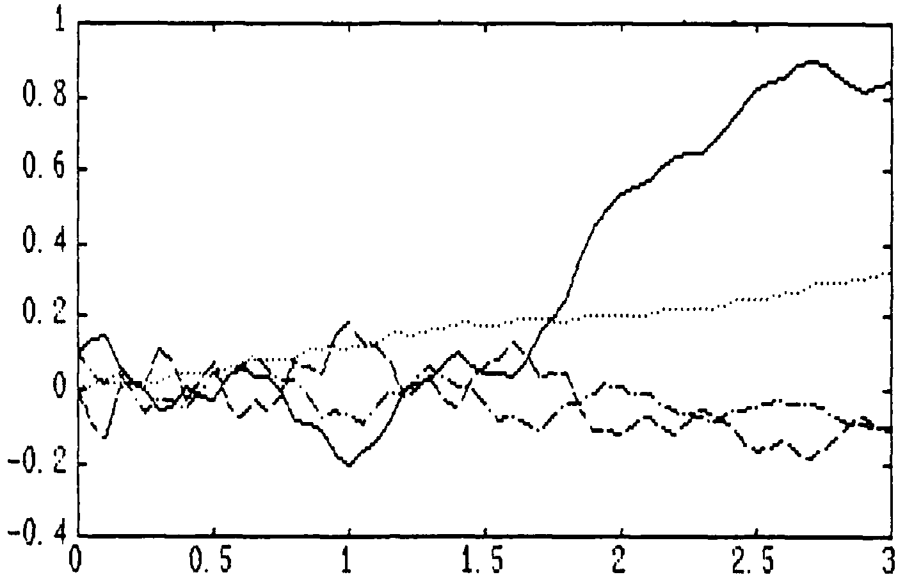


(a) No Damping

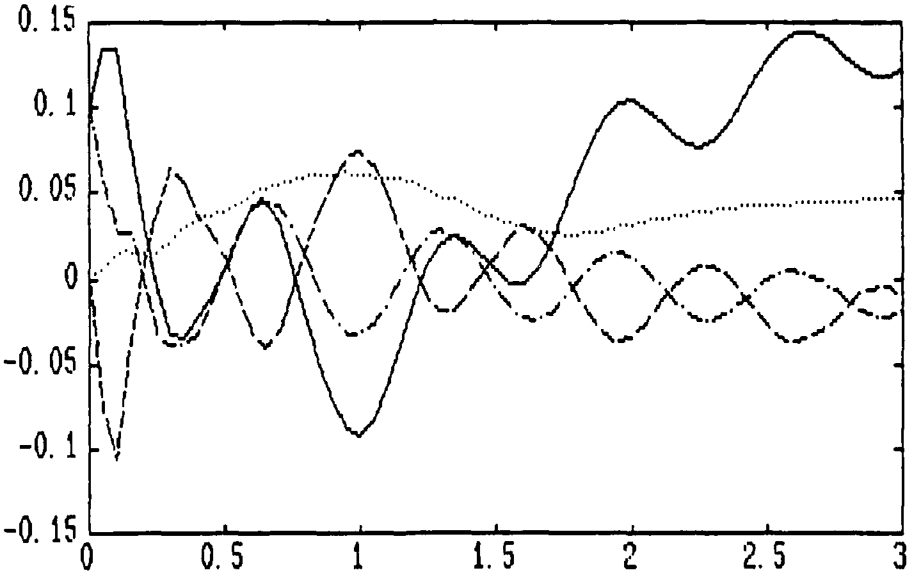
(b) Mild Damping

(c) Moderate Damping

Figure 19. Two Segment Beam Closed Loop Response for Models with Different Damping



(a) Non-Output Responses, Mild Damping



(b) Beam Non-Output Response, NSM=12, L=8, Damping

Figure 20. Effects of Damping on Unmeasured Variables

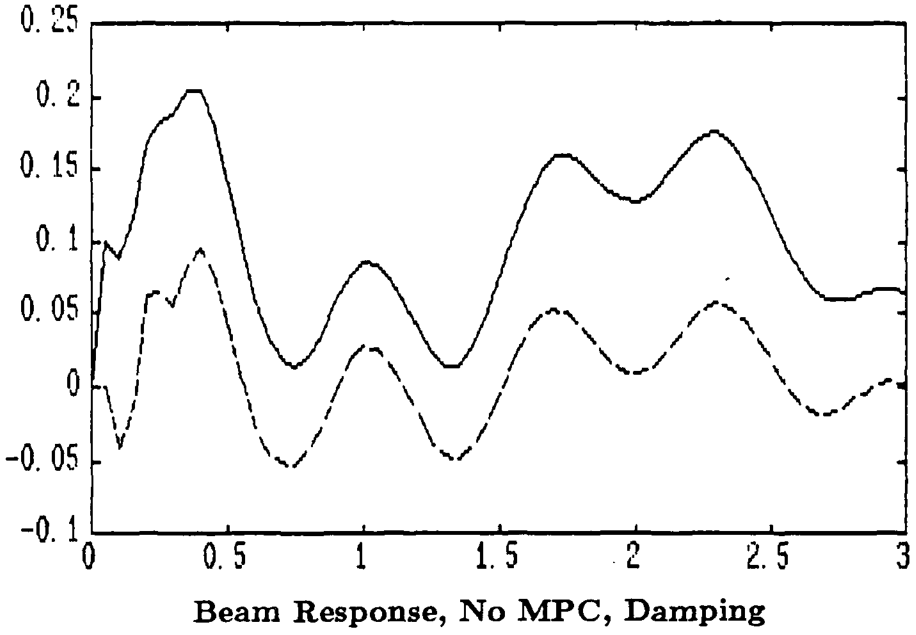
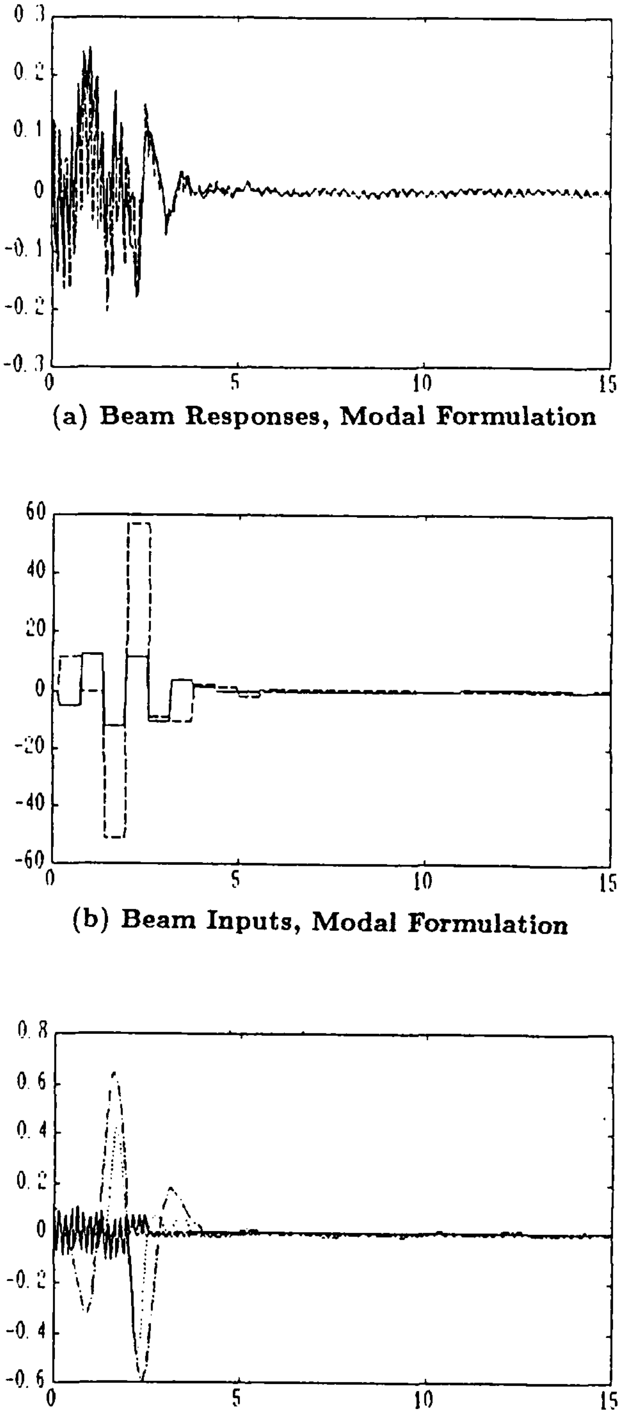


Figure 21. Open Loop Response of Two-Segment Beam with Moderate Damping (Compare with Figure 19c)

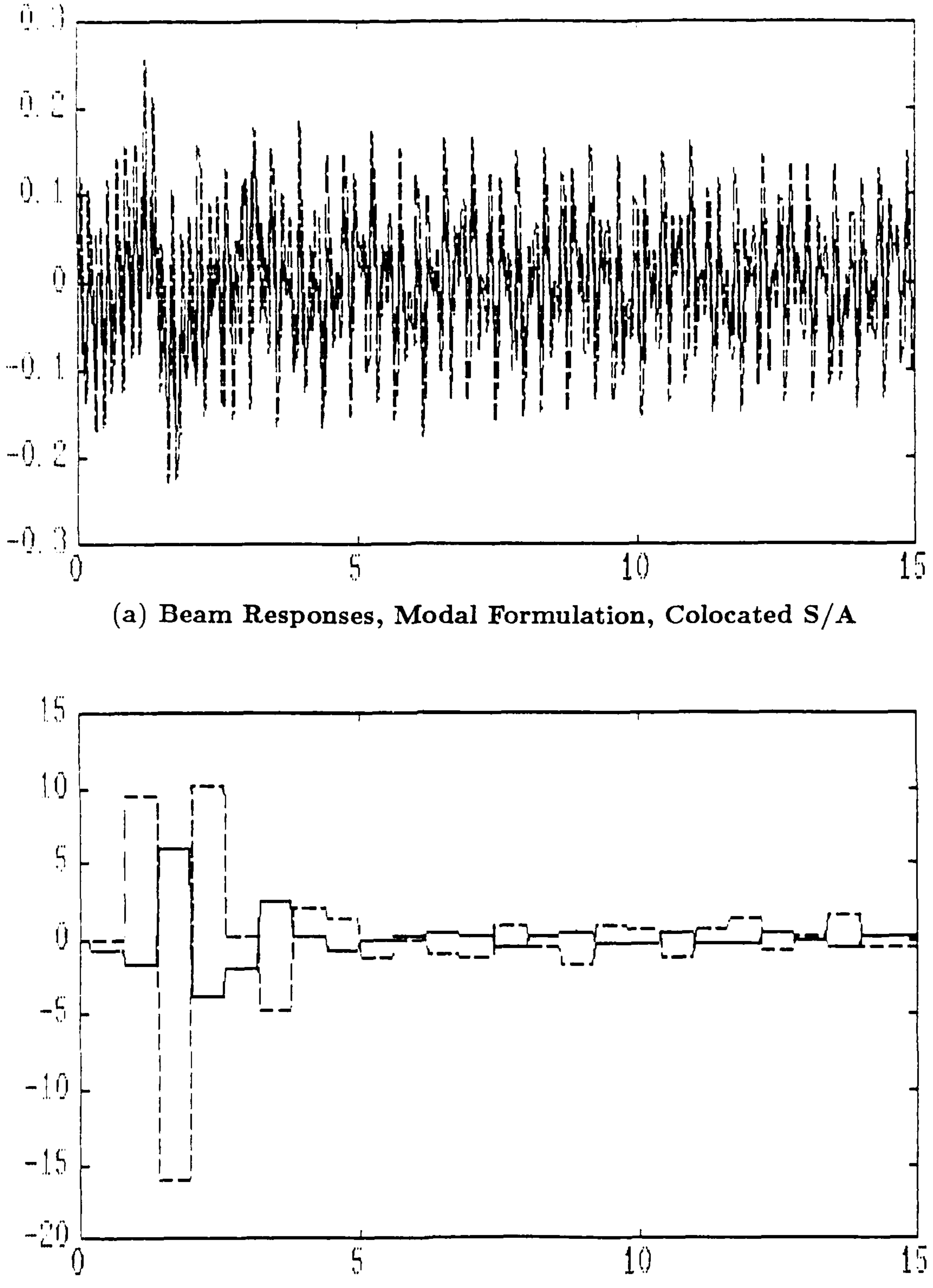


(a) Beam Responses, Modal Formulation

(b) Beam Inputs, Modal Formulation

(c) Beam Non-Output Responses, Modal Formulation

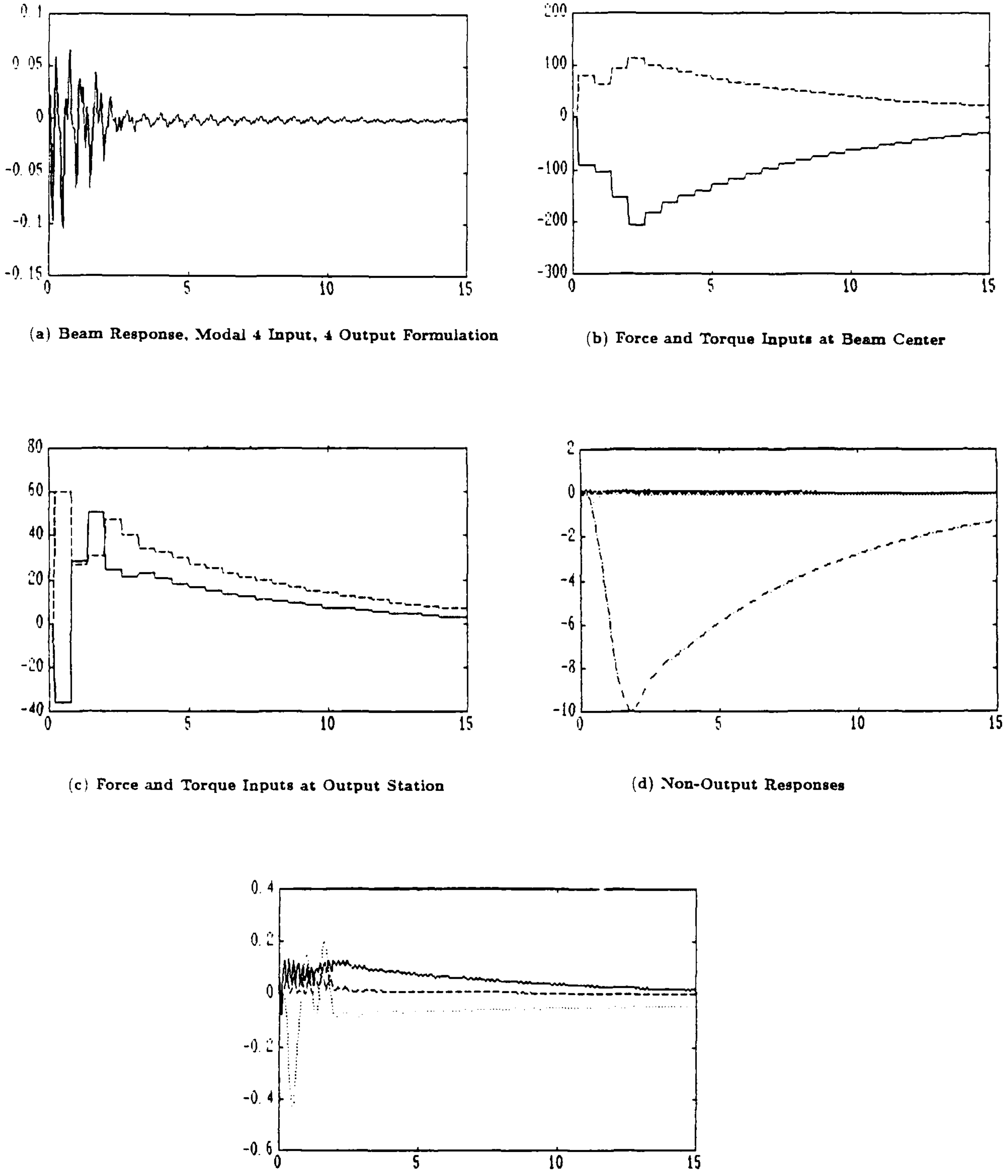
Figure 22. Two Segment Free Beam Closed Loop Response, Using Modal Formulation



(a) Beam Responses, Modal Formulation, Colocated S/A

(b) Beam Inputs, Modal Formulation, Colocated S/A

Figure 23. Two Segment Free Beam Response, Using Colocated Sensors and Actuators



(a) Beam Response, Modal 4 Input, 4 Output Formulation

(b) Force and Torque Inputs at Beam Center

(c) Force and Torque Inputs at Output Station

(d) Non-Output Responses

(e) First 3 Non-Output Responses (Detail)

Figure 24. Two Segment Free Beam response, Using 4 Inputs and 1 Output

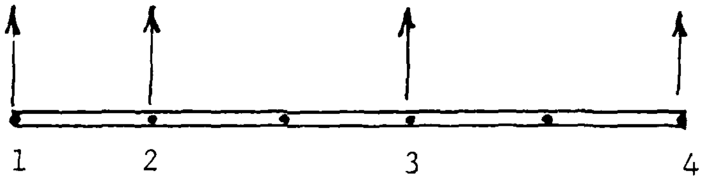
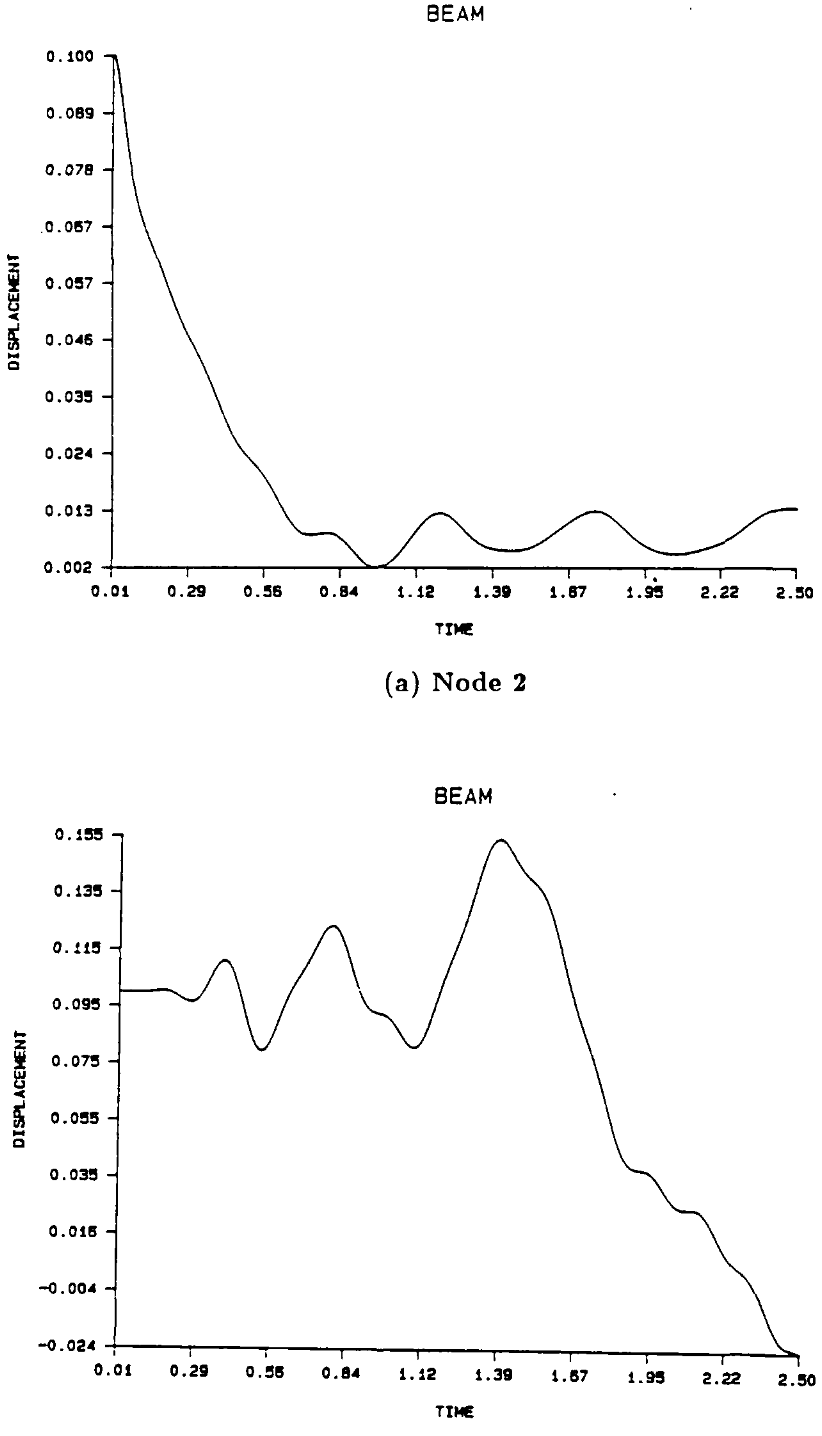


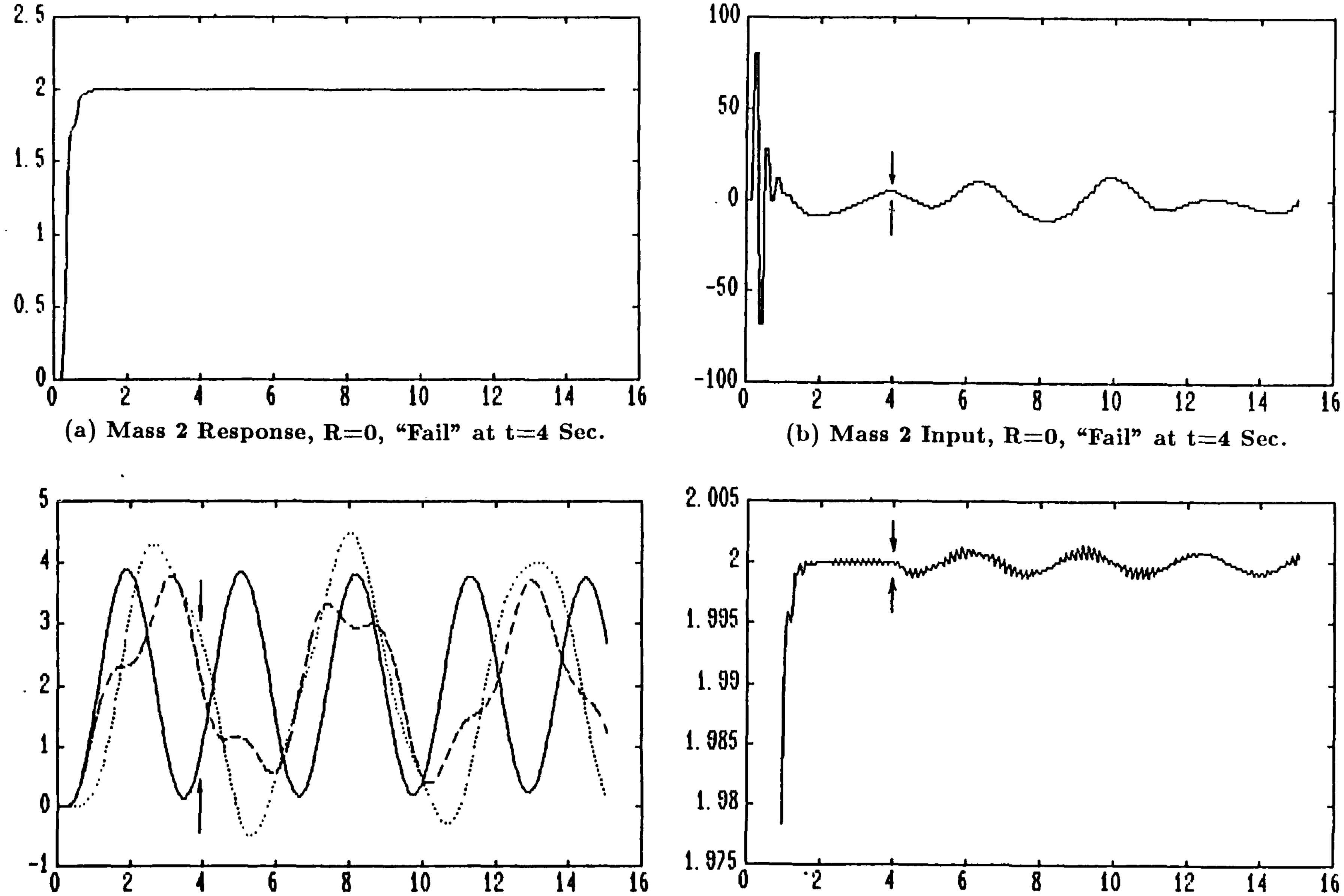
Figure 25. Five-Segment Free-Free Beam Configuration



(a) Node 2

(b) Node 4

Figure 26. Five Segment Free Beam Closed Loop Response Using Preliminary MPC Design



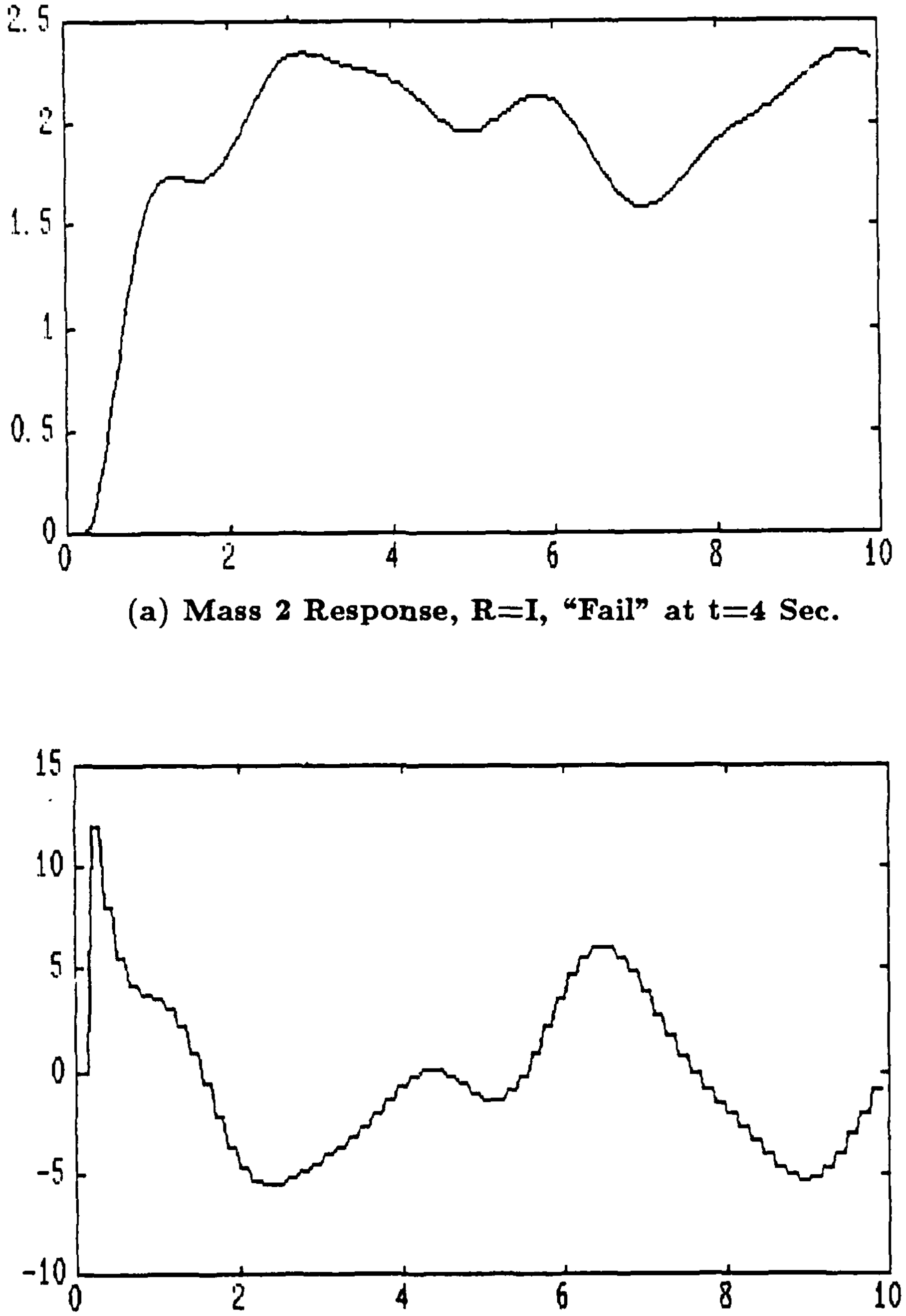
(a) Mass 2 Response, R=0, "Fail" at t=4 Sec.

(b) Mass 2 Input, R=0, "Fail" at t=4 Sec.

(c) Mass 1, 3 and 4 Response, R=0, "Fail" at t=4 Sec.

(d) Mass 2 Response Detail, R=0, "Fail" at t=4 Sec.

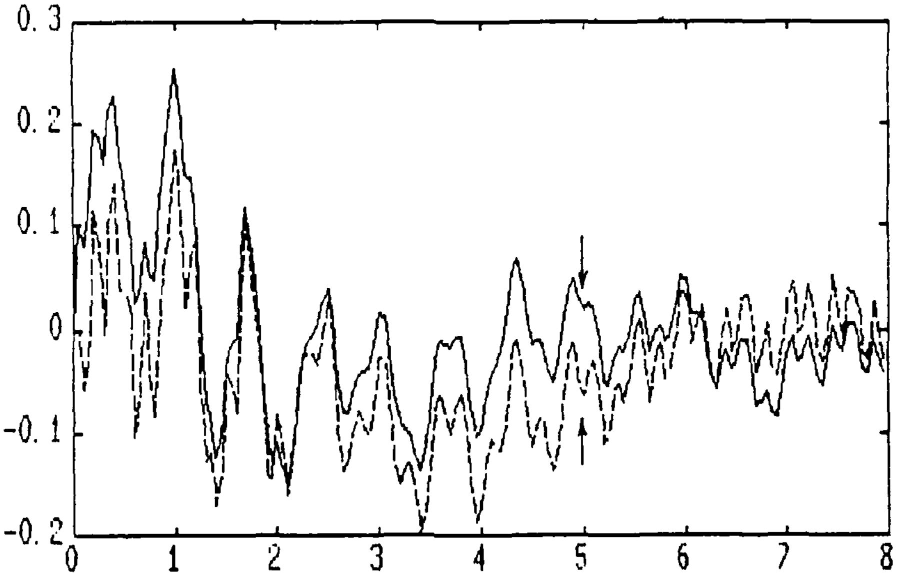
Figure 27. Robustness Analysis of Disk-Torsion Bar System; Damping Reduced by 50% at 4 Seconds



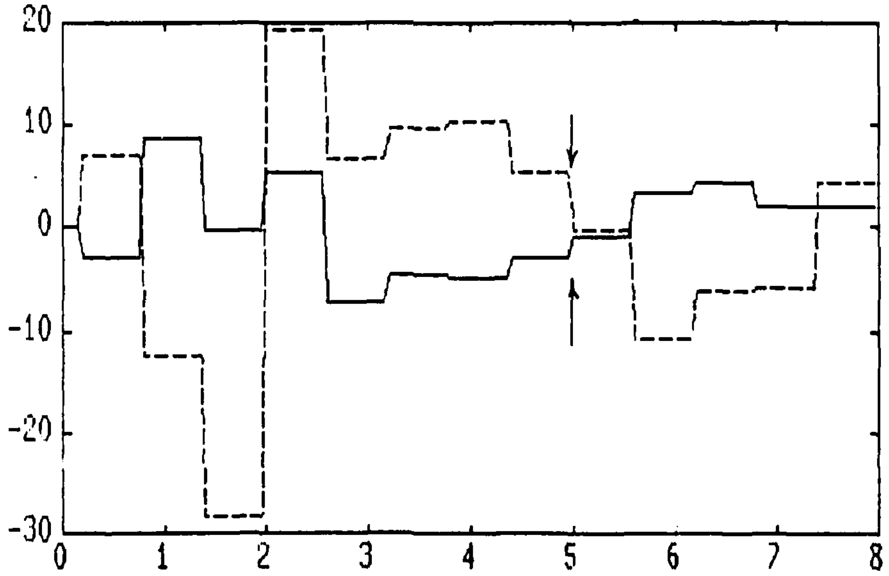
(a) **Mass 2 Response, R=I, "Fail" at t=4 Sec.**

(b) **Mass 2 Input, R=I, "Fail" at t=4 Sec.**

Figure 28. Robustness Analysis of Disk-Torsion Bar System, Showing Effect of Poor Weight Selection



(a) **Beam Response, L=12, 50% More Spring at t=5 Sec.**



(b) **Beam Inputs, L=12, 50% More Spring at t=5 Sec.**

Figure 29. Robus*ι*ness Analysis of Free Beam Controlled by MPC; "Failure" at $t = 5.0$ seconds.

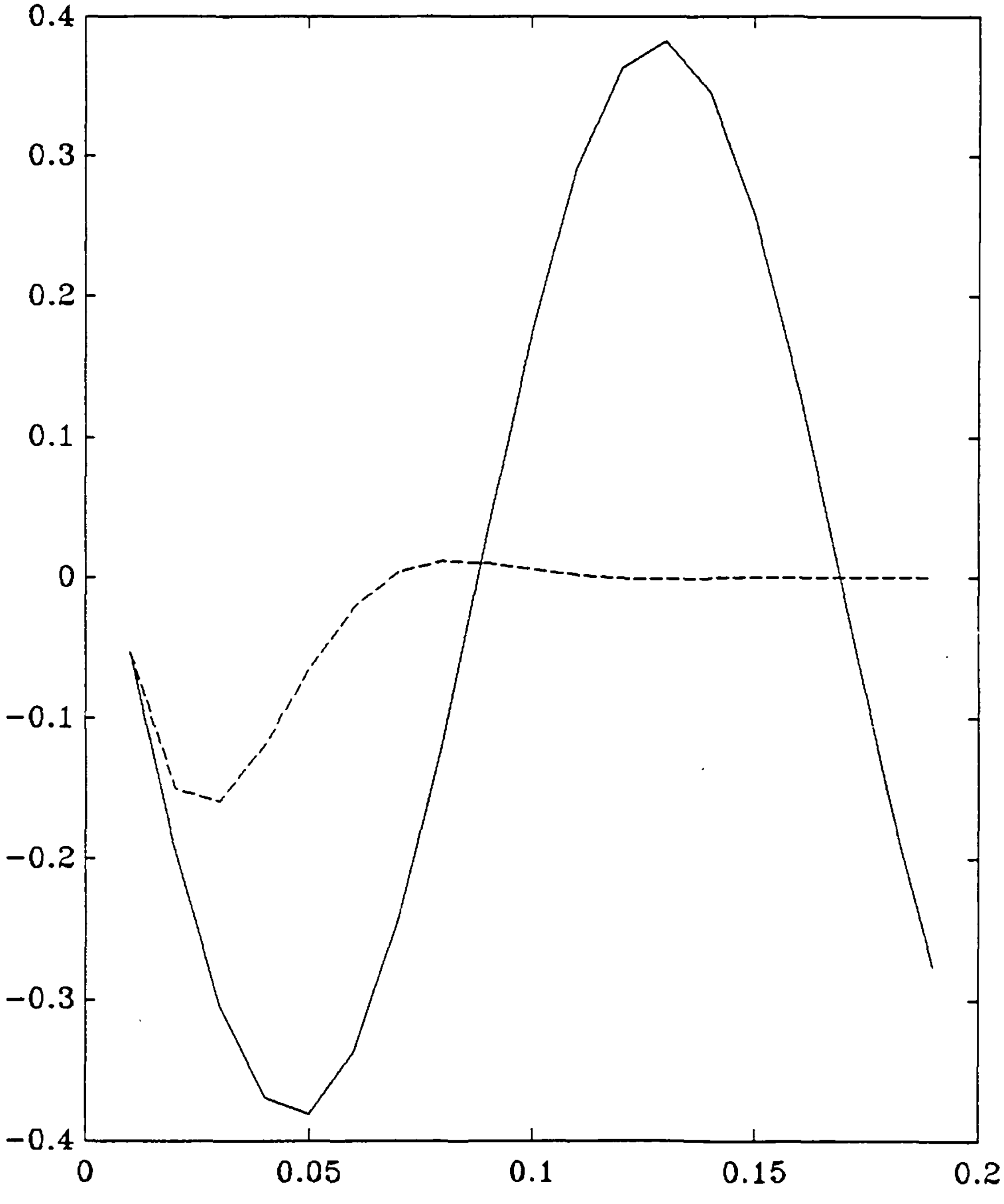


Figure 30. Controlled *vs.* Uncontrolled Modal Respon<sup></sup>ses Using Independent Modal MPC Design Method

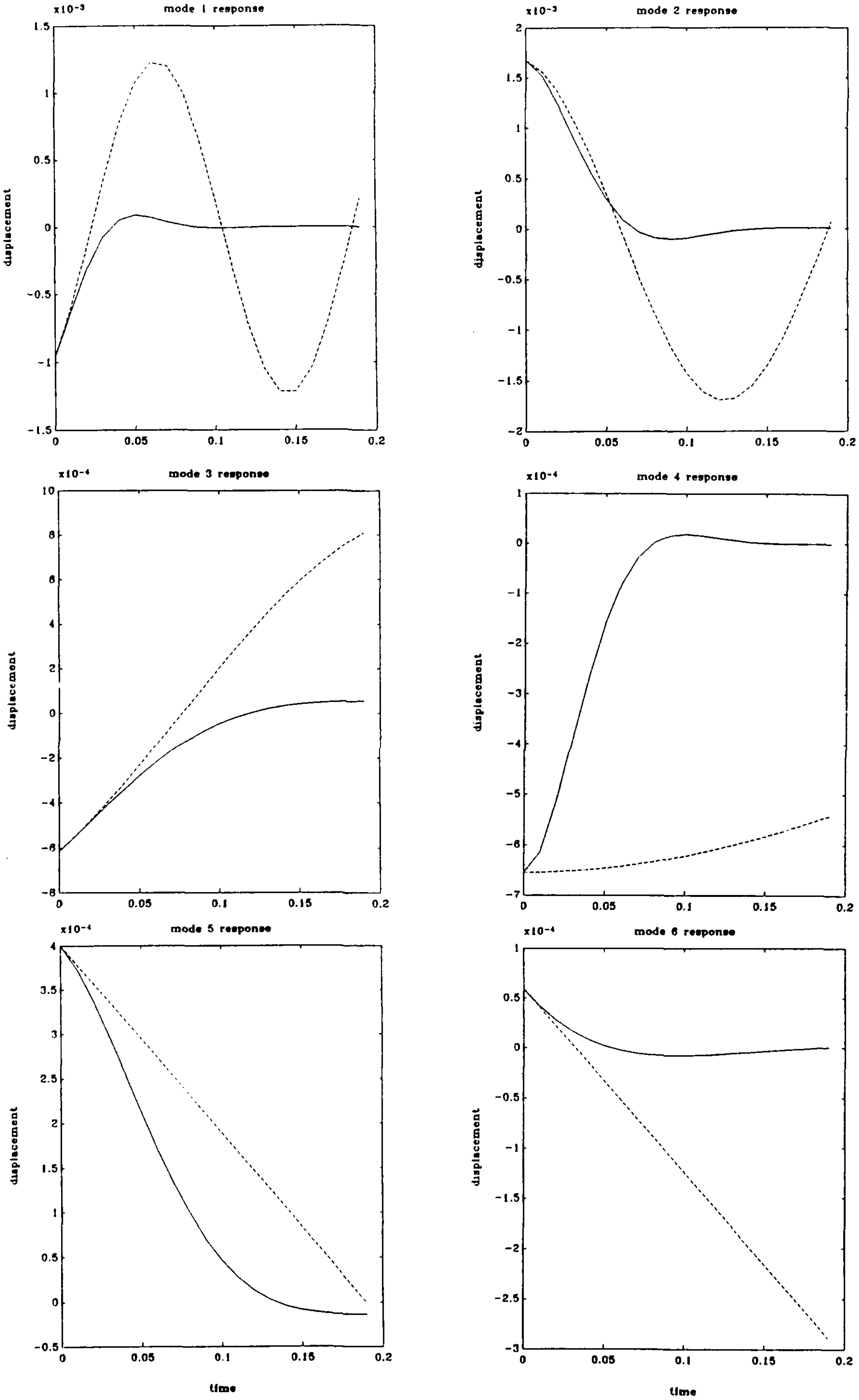


Figure 31. Controlled *vs.* Uncontrolled Modal Responses Using Independent Modal MPC Design Method; All Modes

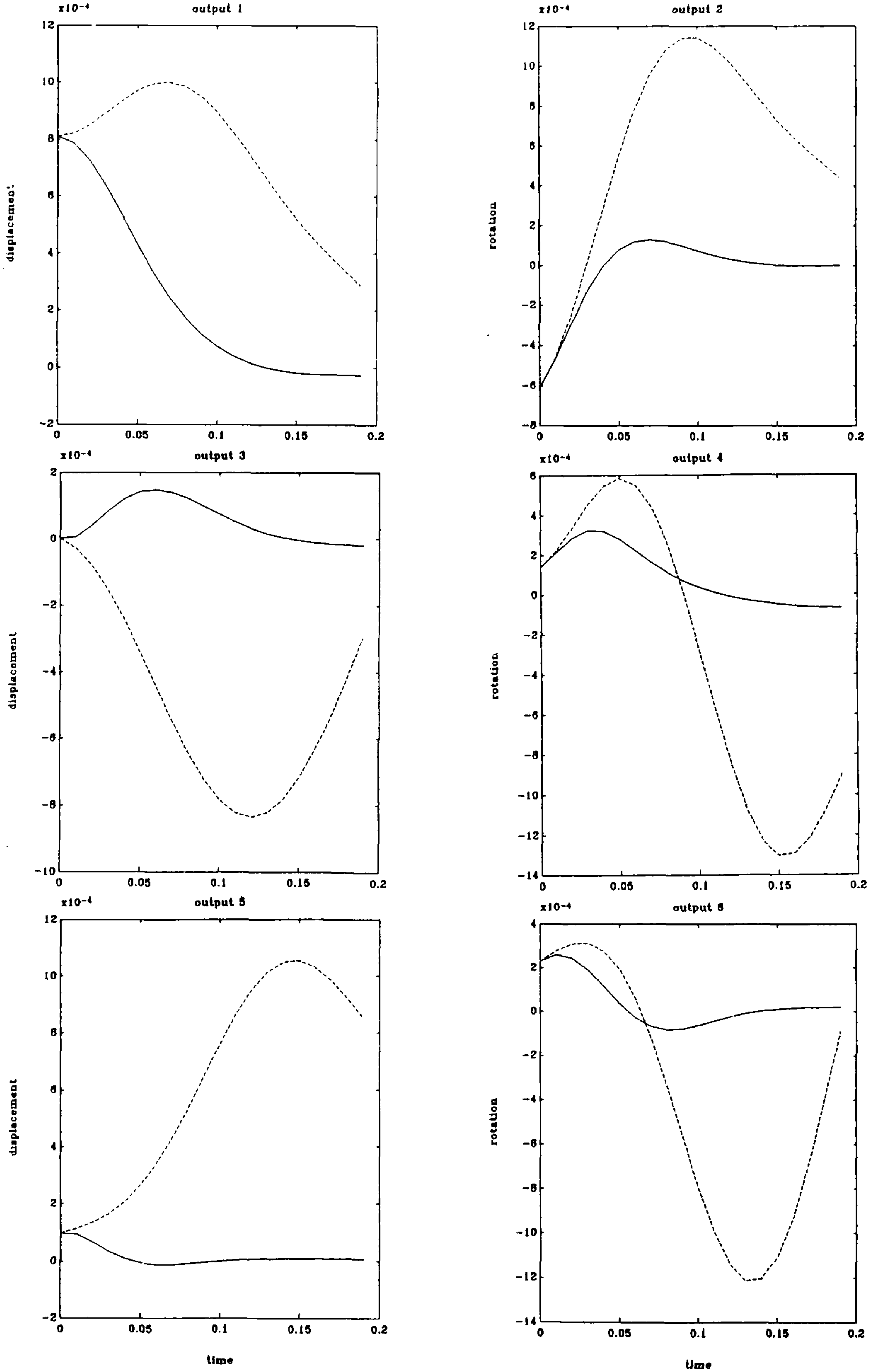


Figure 32. Controlled *vs.* Uncontrolled Physical Responses Using Independent Modal MPC Design Method

# 4 SUMMARY AND CONCLUSIONS

## 4.1 Summary of Results

The results presented above indicate clearly that CVA identification and Model Predictive Control are capable of providing robust control of flexible structures. This conclusion is the result of applying CVA and MPC to three flexible structure models: an 8th order disk-torsion bar system with one input and one output, collocated; a 12th order free-free beam with 2 inputs and 2 outputs; and a 24th order free-free beam with 4 inputs and 4 outputs. The sensors and actuators were not collocated in the 12th order, 2-segment beam problem. Both the smaller and larger beam models were analyzed with basically zero damping, and stabilizing control was achieved. For a sufficient level of input noise and also for the proper level of lags, and some process and/or measurement noise, CVA does well in system identification, although more analysis and tuning of the larger beam case remains to be done.

The analysis to date has established that, for the identification and control of flexible structures, the horizon of prediction must be adequately large. Specifically, $L$ should be at least the order of the system, and so should the number of output smoothing points when the system has minimal or no damping. Both parameters may be reduced somewhat for a structure with as little as 2% damping. Also, it was determined that the system sample period is a sensitive parameter, as is the direct control weight $R$. Assuming adequate control resources, best performance is achieved for $R = 0$. The dimension size of the output error weight $Q$ is very important, but performance thus far is rather insensitive to changes in the values of $Q$ elements.

Sensor/actuator location is important. Deliberately difficult configurations were analyzed in the beam problem, and analysis points to the need for careful consideration of the configuration. MPC stability analysis can assist in this task.

Inherent robustness of LSSICS using MPC was demonstrated, and the potential for its enhancement using CVA identification discussed.

Further robustifying at the algorithm level may result from performing the design in terms of modal coordinates, which decouple the input/output structure, or in terms of the statistically significant dynamic states as determined offline using CVA.

The analysis done in Phase I has confirmed many suppositions but has naturally uncovered or left unresolved other important issues, many of which we propose to address in Phase II.

## 4.2 Conclusions and Recommendations

The work performed under this phase of the project indicates clearly that the combination of CVA identification and adaptive control using MPC offer potential for being a key part of robust LSS control synthesis methods. There are further details to resolve, but results obtained to date indicate that it is worthwhile to pursue their resolution. With regard to the effectiveness of CVA, the conclusion is that some asymmetry is beneficial to some aspects of the overall identification process, but is not critical to achieving good real time control.

Recommendations fall into the area of performing detailed analysis and comparison of this approach with other methods. Metrics related to LSS mission requirements and performance specifications should be developed and incorporated into the design process. More work is needed in fully integrating the CVA and MPC algorithms. Also, further analysis of the sensor and actuator configuration and its effect both on a useful control design and on system performance should be done. Location of actuators and sensors can be combined into the MPC design process using nonlinear programming methods.

The benefits and costs of decentralized control of flexible orbiting structures should also be analyzed, particularly from the standpoint of CVA and MPC applications. Finally, means should be found to incorporate well-understood metrics such as controllability and observability into the selection of key MPC design parameters, e.g., the weighting matrix elements.

# APPENDIX A

## LSS Analysis Software

Grumman Corporation has several programs for large scale control-system synthesis and analysis as well as large-scale structural analysis, which assist in comparison analyses.

**Optimal Control Codes.** Grumman's extensive library of efficient and accurate optimal control and estimation computer codes enable the user to design, analyze, simulate, and evaluate large-scale problems in a relatively short time. The codes reside on permanent libraries on both the batch and time sharing systems so that they are readily accessible. Two major control analysis systems are used.

**DIGISYN.** DIGISYN is a design-oriented program which performs direct analog or digital designs and analyzes the results via covariance analysis, linear time history simulations, frequency response analysis, etc. It allows the user to perform partial state feedback, full state feedback, and optimal estimator designs. DIGISYN is used extensively in aircraft and spacecraft design efforts. Grumman developed this program using corporate funds and considers it proprietary.

**CASCADE.** A second set of codes, this is a library of proprietary Grumman programs, to be used extensively in this study, will be invaluable for large space structures research. Some of the unique features of this package are:

- A built in capability to do normal mode analysis for systems with many degrees of freedom (from several hundred to a thousand)
- A capability to specify the sample time required to control the system based on the overall uncertainty which will be encountered compared with tolerable control system deviations
- A capability to design a continuous or digital control systems which minimizes the same criteria
- A numerical approach which does not require integration to compute equivalent discrete systems and performance indices (all of these depend on the use of a generalized eigenvalue eigenvector program)
- The computation of the optimal full state control gain or Kalman filter gain, in steady state, for continuous or discrete designs which uses the "Potter" approach (again, this is an eigenvalue- eigenvector approach)
- The computation of the "reduced state" feedback gain in continuous or discrete time which solves for the optimal gain using a Davidon minimization of the cost equation
- A capability to design Luenberger observers of any specified dimension and the specification of the minimal observer dimension
- A general controllability-observability analysis which permits sensor/actuator placement so that control authority and measurement information are optimally distributed among the various states being controlled
- A general maximum likelihood or recursive filtering approach to parameter identification. This program is independent of the control system design code that does

all of the previous analysis/synthesis.

CASCADE enables the user to develop the design model directly on the interactive terminal. He specifies the nearest mode in the finite element model that the sensor is mounted, the type of sensor (linear or rotation, absolute or differential), where it is located, and whether it is a force or torque device. It applies forces, the direction that the force is applied. For external disturbances, the user specifies the locations of the disturbance (distributed disturbances appear at many or possibly all nodes on the finite element model). The matrices $A, B, C$ and $D$ in the model:

$$\dot{x} = Ax + Bu \tag{46}$$

$$y = Cx + Du \tag{47}$$

are then determined by CASCADE by reading the appropriate data from a NASTRAN structural-analysis output file.

**Analysis and Simulation.** both linear and nonlinear simulation packages of varying degrees of complexity have been developed at Grumman. The linear package allows time history generation by two methods.

1. Propagation of difference equation

$$x_k = \Phi(\Delta t)x_k + \int_o^{\Delta t} \Phi(\tau)d\tau B u_k \tag{48}$$

$$\Phi(\Delta t) = e^{A\Delta t} \tag{49}$$

where $\Phi$ is the state transition matrix, and the matrices $A$ and $B$ are defined above.

2. Integration of the differential equation between sample times.

**PROTOBLOCK.** PROTOBLOCK is a Grumman-developed program that enables the user to create a control-system block diagram at a computer-graphics workstation. Each block can represent an elementary control element or a higher-order transfer function. Once the block diagram has been created, PROTOBLOCK can automatically generate the state equations as well as time- history graphs of the response. The block diagram can be easily modified and the simulation can be quickly repeated to assess the impact of design changes.

**SATSIM (Satellite Simulation).** This Grumman-developed program enables an extremely detailed simulation of a satellite attitude control system. To facilitate the computation procedure, small relative structural motions are treated in a linear way while nonlinear simulation is employed for large rigid-body motions. Features of this program include:

- Built in capability to handle detailed spacecraft configurations
- Ability to simulate many control system configurations
- Detailed models of actuators and sensors
- Nonlinear modeling
- Accurate computation of expected disturbances
- An automatic NASTRAN interface

SATSIM is in extensive current use to support work for both military and civilian spacecraft design efforts.

**Structural-Analysis Programs.** Grumman has many structural- analysis computer programs; among them are DISCOS, SPACE14, NASTRAN, ASTRAL-COMAP and PLANS. DISCOS (Dynamics Interaction Simulation of Controls and Structure) is designed to analyze the stability of controlled spacecraft which consist of coupled flexible bodies, SPACE14 is a Grumman-generated program for the analysis of controlled rotating satellites. A class of deployment problems, on-board mass motion (e.g., crew members) and fluid motion can be simulated, and environment disturbances (gravity gradient, solar radiation pressure, and aerodynamic loads) are automatically generated. Both the MacNeal-Schwendler version of NASTRAN and Grumman's structural analyzer, COMAP ASTRAL are used extensively for creating and analyzing large- order finite-element models. PLANS (Plastic Analysis of Structures) is a Grumman developed program used for the analysis of structures which can have plastic deformations. It has been used extensively in computerized crash analysis of automotive and aircraft structures.

# 5 REFERENCES

ACOSS-5 (1982), "Active Control of Space Structures," Interim Report RADC-TR-82-21, Lockheed Missiles and Spacecraft Co., March, 1982.

ACOSS-11 (1982), "Active Control of Space Structures," Interim Report RADC-TR-82-295, C.S. Draper Lab., Inc., Cambridge, MA, November, 1982.

ACOSS-12 (1984), "Active Control of Space Structures," Final Report RADC-TR84-28, Lockheed Missiles and Spacecraft Co., February, 1984.

ACOSS-17 (1984), "Active Control of Space Structures," Final Report RADC-TR-84-186, Control Dynamics Co., September, 1984.

ADAMJAN, D.Z. Arov, and M.G. Krein (1971), "Analytic Properties of Schmidt Pairs for a Hankel Operator and the Generalized Schur- Takagi Problem," *Math. USSR Sbornik,* Vol. 15, pp. 31-73.

AKAIKE, H. (1976), "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion," *System Identification: Advances and Case Studies*, R.K. Mehra and D.G. Lainiotis, eds., New York: Academic Press, pp. 27-96.

AKAIKE, H. (1975), "Markovian Representation of Stochastic Processes by Canonical Variables," *SIAM J. Contr.*, Vol. 13, pp. 162-173.

AKAIKE, H. (1974a), "Markovian Representation of Stochastic Processes and its Application to the Analysis of Autoregressive Moving Average Processes," *Ann. Inst. Statistical Mathematics* vol. 26, pp. 363-387.

AKAIKE, H. (1974b), "A New Look at Statistical Model Identification," *IEEE Automatic Control*, Vol. 19, pp. 667-674.

AKAIKE, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in Petrov, B.A. and F. Csaki (eds.), *Intl Symp. on Information Theory*, Akademiai Kiado, Budapest. pp. 267-281.

ANDERSON, T.W. (1958), *An Introduction to Multivariate Statistical Analysis*, New York: Wiley.

ASTROM, K.J. (1987), "Adaptive Feedback Control," *Proc. of IEEE*, Vol. 75, pp. 185-217.

AUBRUN, J.N., et al., (1982), "Active Control of Space Structures," Interim Report RADC-TR-82-21, Lockheed Missiles and Spacecraft Co., March, 1982.

BALAS, M. (1982), "Some Trends in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams," *IEEE Trans. Aut. Contr.*, Vol. AC-27, No. 3, Jun, 1982.

BJORCK, A., and Golub, G.H. (1973), "Numerical methods for computing angles between subspaces," *Numer. Math.*, Vol. 27, pp. 579-94.

BRENT, R., and F. Luk (1985), "The Solution of Singular-Value and Symmetric Eigenvalue Problems on Multiprocessor Arrays," *SIAM J. Scientific and Statistical Computing,* Vol. 6, pp. 69-84.

CAMUTO, E. and Menga, G. (1982), "Approximate Realization of Discrete Time Stochastic Processes," *Sixth IFAC Symposium on Identification and System Parameter Estimation,* Vol. 2, Eds. G.A. Bekey and G.N. Saridis, held June 7-11, 1982, Washington, D.C.

CANDY, J.V., Bullock, T.E., and Warren, M.E. (1979), "Invariant Description of the Stochastic Realization," *Automatica,* Vol. 15, pp. 493-5.

CANNON, Jr.. R.H. and D.E. Rosenthal (1984), "Experiments in control of Flexible Structures with Noncolocated Sensors and Actuators," *J. Guidance and Control*, Sep.-Oct, pp. 546- 553.

CARROLL, J.V., and W.E. Larimore (1986). "Identification and Control of Battle Damaged Aircraft," presented at *NAECON Conference*, Dayton, Ohio.

CARROLL, J.V., S. Mahmood, R. Davis and P. Shaw (1986), "Reconfiguration Strategies for Aircraft Flight Control Systems Subjected to Actuator Failure/Surface Damage," AFWAL Report No. AFWAL-TR-86-3079, December.

COLSON, H.J. (1978), "Application of Model Algorithmic Control to a Lightly Damped Single Input Single Output System," AFIT M.S. Thesis, Air Force Institute of Technology, Wright Patterson AFB, Ohio, December.

CUTLER, C.R. and B.L. Ramaker (1980), "Dynamic Matrix Control - A Computer Control Algorithm," *Proc. Joint Automatic Control Conf.*, paper WP5-B, San Francisco, CA.

DESAI, U.B., and Pal, D. (1984), "A Transformation Approach to Stochastic Model Realization," *Trans. Automat. Control,* Vol. 29, pp. 1097-1100.

GARCIA, C.E., and M. Moyari (1982), "Internal Model Control 1. A Unifying Review and Some New Results," *Ind. Eng. Chem. Proc. Des. Dev.*, Vol 21, pp 308-.

GELFAND, I.M., and Yaglom, A.M. (1959), "Calculation of the Amount of Information About a Random Function Contained in Another Such Function," *Amer. Math. Soc. Trans.*, Series (2), Vol. 12, pp. 199-236 (original *Usp. Mat. Nauk.*, Vol. 12, 3-52, 1956).

GEVERS, M. and Wertz, V. (1982), "On the Problem of Structure Selection for the Identification of Stationary Stochastic Processes," *Sixth IFAC Symposium on Identification and System Parameter Estimation*, Eds. G. Bekey and G. Saridis, Wash. D.C.: Mcgregor Werner, pp. 387-92.

GOLUB, G.H. (1969), "Matrix Decompositions and Statistical Calculations," *Statistical Computation.* R.C. Milton and J.A. Nelder, eds., New York: Academic Press, pp. 365-379.

GRAN, R., Proise, M. (1980), "Flexible Spacecraft Attitude Control - Phase I," Grumman Research Dept. Report RE-605, August.

GRAN, R. and Proise, M. (1982), "Flexible Spacecraft Attitude Control - Phase II Final Report for INTELSAT Contract INTEL- 064," Research and Development Center Report RE-632, August.

GRAN, R. and Proise, M. (1981), "Flexible Spacecraft Attitude Control - Phase II Final Report for INTELSAT Contract INTEl- 064," Research and Development Center Report RE-652, September.

HOTELLING, H. (1936) "Relations between Two Sets of Variates," *Biometrika,* Vol. 28, pp. 321-377.

KAILATH, T. (1974), "A View of Three Decades of Linear Filter Theory," *IEEE Trans. Info. Theory*, Vol. 20, pp. 146-181.

KHATRI, C.G. (1976), "A Note on Multiple and Canonical Correlation for a Singular Covariance Matrix," *Psychometrika*, Vol 41, pp. 465-70.

KOSUT, R.L. and M.G. Lyons (1984), "Issues in the Adaptive Control of Large Space

Structures," presented at 23rd IEEE Conf. on Decision and Control, Las Vegas, NV, December, 1984.

KUNG, S.Y. and D.W. Lin (1981), "Optimal Hankel-Norm Model Reductions: Multivariable Systems," *IEEE Trans. Automatic Control*, vol. 26, pp. 832-852.

LARIMORE, W.E. (1988), "System Identification and Control Using SVDs on Systolic Arrays," *SPIE Symposium on Innovative Science and Technology, Proc. of Conference on High Speed Computing*, Vol. 880, January.

LARIMORE, W.E. (1987). "Identification of Nonlinear Systems Using Canonical Variate Analysis," *Proc. 26th IEEE Conference on Decision & Control*, Los Angeles, CA.

LARIMORE, W.E. (1986), "A Unified View of Reduced Rank Multivariate Prediction Using a Generalized Singular Value Decomposition," Submitted for Publication.

LARIMORE, W.E. and R.K. Mehra (1985), "The Problem of Overfitting Data," *Byte*, Vol. 10, pp. 167-80.

LARIMORE, W.E., S. Mahmood and R.K. Mehra (1984), "Multivariable Adaptive Model Algorithmic Control" *IEEE Conf. on Decision and Control*, Las Vegas, pp. 675-80.

LARIMORE, W.E. (1983a), "Predictive Inference, Sufficiency, Entropy, and An Asymptotic Likelihood Principal," *Biometrika*, Vol. 70, pp. 175-181.

LARIMORE, W.E. (1983b), "System Identification, Reduced-order Filtering and Modeling Via Canonical Variate Analysis," *Proc. Amer. Control Conf.*, San Francisco, pp. 445-451.

LECTIQUE, J., Rault, A., Tessier, M., and Testud, J.L. (1978), "Multivariable Regulation of a Thermal Power Plant Steam Generator," IFAC World Congress, Helsinki.

LUK, F.T. (1985), "A Parallel Method for Computing the Generalized Singular Value Decomposition," *J. Parallel and Distributed Computing,* Vol. 2, pp. 250-260.

MEHRA, R.K., et al. (1978), "Model Algorithmic Control Using IDCOM for the F100 Jet Engine Multivariable Control Design Problem," in *Alternatives for Linear Multivariable Control*, (Ed., Sain et al.), NEC, Chicago.

MOREAU, P., and Littman, J.P. (1978), "European Transonic Wind Tunnel, Dynamics Simplified Model," Adersa/Gerbios.

RAO, C.R. (1965), "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya (A)*, Vol. 26, pp.329-58

RAULT, A., Richalet, J., and LeRoux, P. (1975), "Command Auto Adaptive D'un Avion," Adersa/Gerbios, 75/95.

REID, J.G. (1983), *Linear System Fundamentals, Continuous and Discrete, Classic and Modern*, McGraw-Hill: New York.

REID, J.G., D.E. Chaffin and J.T. Silverthorn (1981), "Output Predictive Algorithmic Control: Precision Tracking with Application to Terrain Following," AIAA Journal of Guidance and Control, Vol. 4, No. 5, Sept.-Oct.

RICHALET, J., Lecamus, F., and Hummel, P. (1970), "New Trends in Identification, Minimization of a Structural Distance, Weak Topology," 2nd IFAC Symposium on Identification, Prague.

RICHALET, J., Rault, A., and Pouliquen, R. (1971), *Identification des Processus par* *la Methode du Modele*, Gorden & Breach.

RICHALET, J., A. Rault, J.L. Testud, and J. Papon (1978), "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica*, Vol. 14, pp. 413-.

STRUNCE, R., et al., (1980), "Active Control of Space Structures: Interim Report," Charles Stark Draper Laboratory Report R-1404, October.

TAHK, M. and J.L. Speyer, (1987), "A Parameter Robust LQG Design Synthesis With Applications to Control of Flexible Structures," *Proc. of the Automatic Control Conference*, Minneapolis, MN, June.

TESTUD, J.L. (1979), "Commande Numerique Multivariable du Ballon de Recuperation de Vapeur," Adersa/Gerbios.

VAN LOAN, C.F. (1976), "Generalizing the singular value decomposition," *SIAM J. Numer. Anal.*, Vol. 13, pp. 76-83.

WAGIE, D.A. and R.E. Skelton (1986), "A Projection Approach to Covariance Equivalent Realizations of Discrete Systems," *IEEE Trans. on Automatic Control*, Vol. AC-31, No. 12, December.

YAGLOM, A.M. (1970), "Outline of Some Topics in Linear Extrapolation of Stationary Random Processes," *Proc. Fifth Berkeley Symp. Math. Stat. and Prob.*, Berkeley, California, California Press, pp. 259-278

END
DATE
10-88
DTIC